KTH Royal Institute of Technology
Universidad del País Vasco

School of Electrical Engineering
and Computer Science

Finite Element simulations: computations and applications to aerodynamics and biomedicine

Massimiliano Leoni

PhD thesis in Computer Science
Supervisor: Prof. Johan Hoffman

School of Electrical Engineering
and Computer Science
TRITA-EECS-AVL-2020:66 KTH Royal Institute of Technology
ISBN 978-91-7873-710-9 Stockholm, SWEDEN

Akademisk avhandling som med tillstånd av Kungl Tekniska högskolan framlägges till offentlig granskning för avläggande av teknologie doktorsexamen i datalogi den 11 december 2020 klockan 14.00 i Kollegiesalen, Kungliga Tekniska Högskolan, Brinellvägen 8, Stockholm.

Tryck: Universitetsservice US AB

Abstract

Partial Differential Equations describe a large number of phenomena of practical interest and their solution usually requires running huge simulations on supercomputing clusters. Especially when dealing with turbulent flows, the cost of such simulations, if approached naively, makes them unfeasible, requiring modelling intervention.

This work is concerned with two main aspects in the field of Computational Sciences. On the one hand we explore new directions in turbulence modelling and simulation of turbulent flows; we use an adaptive Finite Element Method and an *infinite Reynolds number* model to reduce the computational cost of otherwise intractable simulations, showing that we are able to perform time-dependent computations of turbulent flows at very high Reynolds numbers, considered the main challenge in modern aerodynamics.

The other focus of this work is on biomedical applications. We develop a computational model for (Cardiac) Radiofrequency Ablation, a popular clinical procedure administered to treat a variety of conditions, including arrhythmia. Our model improves on the state of the art in several ways, most notably addressing the critical issue of accurately approximating the geometry of the configuration, which proves indispensable to correctly reproduce the physics of the phenomenon.

Resumen

Las ecuaciones en derivadas parciales describen muchos fenómenos de interés práctico y sus soluciones suelen necesitar correr simulaciones muy costosas en clústers de cálculo. En el ámbito de los flujos turbulentos, en particular, el coste de las simulaciones es demasiado grande si se utilizan métodos básicos, por eso es necesario modelizar el sistema.

Esta tesis doctoral trata principalmente de dos temas en Cálculo Científico. Por un lado, estudiamos nuevos desarrollos en la modelización y simulación de flujos turbulentos; utilizamos un Método de Elementos Finitos adaptativo y un modelo de *número de Reynolds infinito* para reducir el coste computacional de simulaciones que, sin estas modificaciones, serían demasiado costosas. De esta manera conseguimos lograr simulaciones evolutivas de flujos turbulentos con número de Reynolds muy grande, lo cual se considera uno de los mayores retos en aerodinámica.

El otro pilar de esta tesis es una aplicación biomédica. Desarrollamos un modelo computacional de Ablación (Cardiaca) por Radiofrecuencia, una terapia común para tratar varias enfermedades, por ejemplo algunas arritmias. Nuestro modelo mejora los modelos existentes en varias maneras, y en particular en tratar de obtener una aproximación fiel de la geometría del sistema, lo cual se descubre ser crítico para simular correctamente la física del fenómeno.

Sammanfattning

Partiella differentialekvationer kan användas för att beskriva ett stort antal fenomen av praktiskt intresse. Vanligtvis krävs enorma simuleringar på superdatorkluster för att hitta deras lösningar. I synnerhet vid arbete med turbulent flöde. Dessa simuleringar är så resurskrävande att utan specialbehandling så är de ohanterbara och kräver manuella modelleringsingrepp.

Denna avhandling består av två huvuddelar. Först utforskar vi nya riktningar i turbulensmodellering och simulering av turbulent flöde. Vi använder oss av en adaptiv finit elementmetod och en modell med oändliga *Reynoldstal* för att reducera beräkningskostnaden för annars ohanterbara simuleringar. Avhandlingen visar att vi lyckats utföra tidsberoende beräkningar av turbulent flöde vid väldigt höga Reynoldstal, vilket är en av de stora utmaningarna i modern aerodynamik.

Den andra delen i denna avhandlingen fokuserar på biomedicinska tillämpningar. Vi har utvecklat en modell för radiofrekvensablation, ett populärt medicinskt ingrepp som är del i behandlingen av ett flertal sjukdomar, inklusive arytmi. Vår modell överträffar befintliga modeller på flera punkter. Mest markant genom att noggrant approximera konfigurationens geometri, vilket är väsentligt för att korrekt kunna reproducera fenomenets fysik.

Preface

This thesis consists of an introduction and six papers.

Paper I

V.-d. Nguyen, M. Leoni, T. Dancheva, et al. "Portable simulation framework for diffusion MRI". in: *Journal of Magnetic Resonance* 309 (Dec. 2019), p. 106611

Paper II

J. Jansson, E. Krishnasamy, M. Leoni, et al. "Time-resolved adaptive direct fem simulation of high-lift aircraft configurations". In: *Numerical Simulation of the Aerodynamics of High-Lift Configurations*. Cham: Springer International Publishing, 2018, pp. 67–92

Paper III

A. Petras, M. Leoni, J. M. Guerra, et al. "A computational model of open-irrigated radiofrequency catheter ablation accounting for mechanical properties of the cardiac tissue". In: *International Journal for Numerical Methods in Biomedical Engineering* 35.11 (Nov. 2019)

Paper IV

A. Petras, M. Leoni, J. M. Guerra, et al. "Tissue Drives Lesion: Computational Evidence of Interspecies Variability in Cardiac Radiofrequency Abla-

tion". In: *Functional Imaging and Modeling of the Heart*. 2019, pp. 139–146

Paper V

A. Petras, M. Leoni, J. Guerra, et al. "Effect of Tissue Elasticity in Cardiac Radiofrequency Catheter Ablation Models". In: *2018 Computing in Cardiology Conference (CinC)*. vol. 2018-Septe. IEEE, Dec. 2018, pp. 2018–2021

Paper VI

Petras, A., Weidmann, Z. M., Leoni, M., Guerra, J. M. and Gerardo-Giorda, L. "Systematic characterization of High-Power Short-Duration Ablation: Insight from an advanced virtual model." *Submitted*.

Contributions and division of work

In complex research projects, results are very often the result of the collaboration of many people. This section aims at clarifying my contributions to the scientific output listed above. As explained in more detail in the following pages, the scope of this thesis is the development of a high-performance computing platform for Finite Element simulations. The major application discussed in this thesis is Cardiac Radiofrequency Ablation, but the methods and software I developed are general and, in principle, easy to apply to a variety of different problems. Paper I reflects this statement by showing an implementation of a Finite Element simulation platform for a different biomedical application, Diffusion Magneto-Resonance Imaging. This paper was also part of its first author's doctoral thesis, but my contributions were complementary, concerning primarily the development and deployment of the containers and the writing of the corresponding sections in the final manuscript, in accordance with my research interests. Some of these contributions were made possible thanks to the experience I gained while taking part to the MSO4SC project, a European project focused on similar technologies – which unfortunately did not produce scientific output suitable for inclusion in this thesis.

Paper II, apart from being an interesting scientific result in itself, serves as an important validation case for the technology developed during my doctoral project. The application, simulation of flow past an aeroplane, pushes the limits of what simulations in this area can achieve. These results required a

significant amount of effort in terms of software development, which I single-handedly took care of. Alongside code development, I also performed some simulations and wrote parts of the manuscript.

Papers III-VI concern Cardiac Radiofrequency Ablation. These are the main results of this thesis, and can be divided into two groups. Paper III contains more fundamental scientific investigations and results, consisting of the model that we developed and the software implementation that brings the model to (virtual) life. On the software development side, I contributed the entirety of the processing code and the lion's share of the pre- and post-processing code. On the modelling side, the first author, the last author and I contributed in comparable amounts to the final model. The same people also wrote the manuscript, to which I contributed the technical parts related to my contributions outlined above.

Papers IV-VI describe self-contained studies and investigations that were made possible by the code and model developed in Paper III. These results focus on the modelling and physical understanding of the phenomena involved in Cardiac Radiofrequency Ablation, with the code that I developed being the cornerstone of this activity. My contributions included adding functionalities to the code to make the relevant simulations possible, supporting my collaborators in using the code, running some test simulations and reviewing the corresponding manuscripts.

Contents

Part I

Introductory chapters

0.1 Aim and scope of this thesis

This thesis concerns the development of methods and software in high-performance Finite Element computations, with a focus on applications, in particular to computational mechanics and biomedicine. In other words, the goal is to develop methods and software that can be used to reproduce, with a computer simulation, scenarios of scientific or technical interest, delivering them to the realm of domain sciences such as biology and medicine, which are at the forefront of impactful scientific discovery. The overall aim is to create tools for domain scientists to use to be able to better advance their own research, exploiting the fact that mathematics lets us abstract away much of the details and tackle the fundamental issues directly.

Accordingly, the following chapters will describe how, in general, our software can be made easily available to the people who can benefit the most from using it and how it can efficiently simulate very complex scenarios, considered unfeasible until recently. Later on, I will present a concrete example of a novelty-rich model that describes a complex, multi-physics setting of great practical importance alongisde its accompanying software implementation. The relevance of these results is twofold: on the one hand, we deliver tools to boldly simulate what no code has simulated before, while on the other hand we bring several related research questions within easy reach.

Chapter 1

Introduction

I have always been interested in mathematics, and I have always been fascinated by science. I find it obvious and surprising, at the same time, that I found myself steering my own research focus in the direction of applications of mathematics to problems in science and technology. The field of numerical simulations offers the perfect ground to grow and nurture my passions into real scientific progress, being at the intersection of mathematics, a variety of applied sciences, and computer programming – a dark art I find difficult to resist. This is the gist of this thesis: I will take my reader through advances in simulation technology and applications to challenging problems in aeronautical engineering and biomedicine, showing how the same techniques can be seamlessly applied to these very different fields of study, in an attempt to celebrate how Mathematics truly is *the art of giving the same name to different things*. Let us dive in.

Given the wide range of topics this thesis necessarily covers, the present chapter briefly recalls the fundamental ideas of the building blocks upon which later chapters are built. This is intended to give a high-level understanding of the concepts to readers coming from different backgrounds and does not aim to be a complete introduction to the corresponding topics. I assume only basic knowledge of calculus, mechanics and computer science on the reader's side.

1.1 The Finite Element Method

The Finite Element Method, or FEM for brevity, is a technique to approximate the solution to partial differential equations, or PDEs. This method is very popular in many fields of engineering due to its versatility and its ability to gracefully handle complex geometries, typical of practical applications [21]. Part of its success is due to the fact that it has a very solid theoretical background, originating in Functional Analysis [4, 22], that clearly identifies problems for which this method is a safe choice. Unlike other popular discretisation methods, in fact, it is relatively easy to prove that the Finite Element Method does yield the correct solution [20].

I will now briefly illustrate the principles of the Finite Element Method on the very famous Poisson equation

$$\begin{cases} -\Delta u = f, & \text{in } \Omega \\ u = 0, & \text{on } \Gamma_D \\ \partial_n u = 0, & \text{on } \Gamma_N \end{cases} \tag{1.1}$$

where $\Omega \subset \mathbb{R}^n$ is an open, connected, regular domain, Γ_D and Γ_N are two disjoint subsets of $\partial\Omega$, such that $\Gamma_D \cup \Gamma_N = \partial\Omega$, where Dirichlet and Neumann boundary conditions are applied, respectively. Here, $u = u(\mathbf{x})$ and $f = f(\mathbf{x})$ are functions on Ω. Considering non-homogeneous Dirichlet and Neumann boundary conditions just adds inessential details to the following discussion without changing the ideas, so we will not be concerned with it here[1]. Finally, $\Delta = \operatorname{div} \nabla$ is the Laplace operator and ∂_n is the normal derivative operator, defined as $\partial_n u = \nabla u \cdot \mathbf{n}$ with $\mathbf{n}$ the unit normal.

Some background is required before jumping into the FEM formulation of Equation (1.1). Equation (1.1), in its classical interpretation, means that we are looking for a function u that satisfies it in every point $\mathbf{x}$ in the closure of the domain, $\overline{\Omega}$. Such a function needs to be twice continuously differentiable in Ω and continuous up to the boundary, or $u \in C^2(\Omega) \cap C^0(\overline{\Omega})$ for short, otherwise it would not make sense to talk about the Laplacian of u.

This requirement turns out to be too strict for many practical applications, resulting in problems that do not have a solution but that try to describe phenomena that do exist. A way to overcome this is to relax the above requirement by choosing a test function v in an appropriate function space X, multiplying Equation (1.1) by it and integrating by parts. Keeping in mind

[1] Basically, one can reduce non-homogeneous boundary conditions to homogeneous ones by an appropriate change of variables.

that, for a scalar function w and a vector function ψ, the Leibniz rule reads

$$\operatorname{div}(w\psi) = w \operatorname{div} \psi + \psi \cdot \nabla w \tag{1.2}$$

and using the Stokes theorem

$$\int_\Omega \operatorname{div} \psi = \int_{\partial\Omega} \psi \cdot \mathbf{n} \tag{1.3}$$

Equation (1.1) can be rewritten as

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v + \int_{\partial\Omega} \partial_n u \; v. \tag{1.4}$$

The improvement is that this equation now makes sense under less strict conditions, namely that the integrals it features converge. We can recover the boundary conditions in Equation (1.1) by modifying the boundary term in Equation (1.4), splitting it into two integrals over Γ_D and Γ_N, substituting $\partial_n u = 0$ in the latter, which disappears, and requiring the test function v to be zero on Γ_D, making the first term disappear as well. The final result is

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v. \tag{1.5}$$

We can now claim that a function u satisfies Equation (1.1) if it satisfies Equation (1.5) for all test functions $v \in X$. The degree to which this claim is correct depends on wether the chosen space X is sufficiently large. We note that the opposite claim is obviously true. The choice of X requires some knowledge of Functional Analysis; in this text I will only report that a natural choice is $H^1_{\Gamma_D}(\Omega)$, the set of functions u that satisfy

$$\int_\Omega u^2 < +\infty \qquad \text{and} \qquad \int_\Omega \|\nabla u\|^2 < +\infty$$

and that are zero[2] on Γ_D. With this choice, it is also natural to look for a solution u in the same space.

We finally have what is called a *variational formulation* of Equation (1.1): find $u \in H^1_{\Gamma_D}(\Omega)$ such that

$$\int_\Omega \nabla u \cdot \nabla v = \int_\Omega f v \qquad \forall v \in H^1_{\Gamma_D}(\Omega). \tag{1.6}$$

[2] In an appropriate sense.

The left-hand side and the right-hand side of Equation (1.6) define a bilinear form $a : H^1_{\Gamma_D}(\Omega) \times H^1_{\Gamma_D}(\Omega) \to \mathbb{R}$ and a linear form $L : H^1_{\Gamma_D}(\Omega) \to \mathbb{R}$, respectively. The equation is often expressed in the more compact form: find $u \in H^1_{\Gamma_D}(\Omega)$ such that

$$a(u, v) = L(v) \qquad \forall v \in H^1_{\Gamma_D}(\Omega). \tag{1.7}$$

We are now ready to give a FEM formulation of Equation (1.6). The idea is to define a finite dimensional approximation of X, which is an infinite dimensional space, so that we can enter the computation in a computer and get an output. Given a simplicial mesh $\mathcal{T}_h$ – a mesh made of triangles or tetrahedra – describing Ω, we can use, for example, the space $V_h = \mathbb{P}^1$ of functions that are linear on every cell of $\mathcal{T}_h$ and globally continuous (and incorporate the Dirichlet boundary condition). The dimension of this space is the number N of the vertexes in the mesh, so it is finite dimensional.

Applying this approximation, our problem now reads: find $u_h \in V_h$ such that

$$\int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega f v_h \qquad \forall v_h \in V_h \tag{1.8}$$

The final step to make the solution computable is to observe that Equation (1.8) is true if and only if it is true for a set of functions $\{\varphi_i\}_i$ that form a basis of V_h. This is the case because, if it is true for a set of basis functions, then we have that

$$\begin{aligned}\int_\Omega \nabla u_h \cdot \nabla v_h = \int_\Omega \nabla u_h \cdot \nabla \left(\sum_i v_h^i \varphi_i \right) &\overset{\text{lin}}{=} \sum_i v_h^i \int_\Omega \nabla u_h \cdot \nabla \varphi_i \\ &\overset{\text{hyp}}{=} \sum_i v_h^i \int_\Omega f \varphi_i \overset{\text{lin}}{=} \int_\Omega f \left(\sum_i v_h^i \varphi_i \right) = \int_\Omega f v_h. \end{aligned} \tag{1.9}$$

The opposite implication is proved immediately by taking $v_h = \varphi_i$.

Incidentally, being $u_h \in V_h$, it can also be expressed as a linear combination of the functions $\{\varphi_i\}_i$ with unknown coefficients $\{u_h^i\}_i$, called *degrees of freedom* in this context:

$$u_h = \sum_{i=1}^N u_h^i \varphi_i.$$

Putting it all together, Equation (1.8) becomes

$$\sum_{i=1}^N u_h^i \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j = \int_\Omega f \varphi_j \qquad \text{for } j = 1, 2, \ldots, N, \tag{1.10}$$

which can be interpreted as a linear system

$$A\mathbf{u} = \mathbf{b}$$

with

$$A_{ij} = \int_\Omega \nabla \varphi_i \cdot \nabla \varphi_j, \qquad \mathbf{u}_i = u_h^i, \qquad \mathbf{b}_i = \int_\Omega f\varphi_i.$$

The solution to this linear system contains the coefficients of the expansion of the unknown u_h as a linear combination of basis the functions $\{\varphi_i\}_i$, hence the solution to the approximated problem.

The above is only a bird's eye view of how the Finite Element Method is used to approximate the solution to PDEs and skips most of the details and issues one runs into both on the theoretical side, such as existence, uniqueness and stability of a solution, and on the practical side, such as conditioning of the linear system. The interested reader can find more details on these topics in Evans [4], Quarteroni [20], and Salsa [22].

1.2 Goal-oriented mesh adaptivity

One of the biggest issues one encounters, in the practice of FEM simulations, is that the linear system obtained from the spatial discretisation can be very big. This happens, for example, if the mesh used for the simulation has a lot of cells, or if the approximating space contains high-order polynomials, both choices that induce a large number of degrees of freedom. Intuitively, the bigger the number of the degrees of freedom, the higher the accuracy of the numerical solution: for an increased cost we expect an increased return. Unfortunately, this can be taken to a point where the computational cost to solve the linear system exceeds that of a regular workstation.

Several approaches are possible to overcome this issue. One of them is to run the simulation on a parallel supercomputing cluster; this is common practice in modern computational science, but even today's gigantic machines struggle to run huge simulations efficiently sometimes. The fact of the matter is that the underlying problem can be so computationally expensive that even parallel algorithms that scale near perfectly are not enough.

This does not at all mean that parallel computing can be disposed of: it is, of course, an indispensable tool and our research, and not only ours, would be impossible without it. What this means is that this approach can greatly benefit from integration with other techniques, such as adaptive computations.

The idea behind adaptive computing is that of optimising the number of degrees of freedom in order to run a simulation with a target quality without

spending more than what is necessary to achieve said quality [20]. As an example, consider the simulation of flow past a cylinder, a classical example in fluid dynamics. When it comes to creating a mesh for the domain, a naive approach is to fix a maximum size h_M and make a mesh whose biggest triangle's diameter is less than h_M. The resulting mesh will have many degrees of freedom close to the cylinder, where we *intuitively* expect to see interesting phenomena, and maybe a Von Kármán's vortex street, but also many degrees of freedom in areas of the domain far from the cylinder, where nothing is happening. If we are simulating the flow of, say, water past a pillar holding a bridge, typically we are doing so because we are interested in extracting some meaning from the simulation, such as the lift force that water exerts on the pillar, to ensure it will not make the structure resonate near its characteristic frequency. Intuitively, we care about high accuracy in the numerical solution where that accuracy pays off in terms of the quantities we are interested in, but we care less about all the small eddies far from the pillar. All degrees of freedom are equal, but some are more equal: we would like to only pay for degrees of freedom that actively contribute to extracting knowledge from the simulation.

Since the solution is unknown, it is difficult to know where, in the domain, more accuracy in the solution will be beneficial. One way is to make an educated guess from experience: professionals with several years of expertise in simulating a certain class of phenomena can, to some extent, guess what parts of a domain are more critical, but this is hardly a reliable technique in general and difficult to apply to new scenarios.

There is a different approach to mesh adaptivity that relies entirely on computation and is one of the main features of the present work: *a posteriori* adaptivity. As the name suggests, this technique extracts information on where to refine a mesh from the solution on a previous mesh. It is to be used as follows: starting from an initial coarse mesh, one can compute a cheap solution, then look at its features and refine the mesh where more accuracy is beneficial, then repeat until some criterion is satisfied. The stopping criterion can be the computational cost, the minimum mesh size, or even something more meaningful such as an estimate of the approximation error, if available.

In some cases, such as the flow past a cylinder described above, one can choose to stop once an estimate of the error on the drag is below a certain threshold. Clearly, the choice of the criterion profoundly influences the adaptivity procedure: trying to minimise the approximation error on the solution itself, one goes back to asking for all the eddies, whirls and details in the flow field; trying to minimise the error on a specific functional, like the lift or the

drag, on the other hand, means conceding that we accept more error on the solution to the equation where that error does not spoil the quantity we are really interested into.

The final piece of the puzzle is how to obtain an error estimate on a functional of the solution, such as the drag. One way to do it, and the one we use in our flow applications, is based on solving the dual problem of the given problem; this technique goes under the name of *dual-based* adaptivity. The dual problem of Equation (1.7) features the bilinear form a^* defined by $a^*(u, v) = a(v, u)$. Without going into too much detail, the solution to the dual problem: find $\psi \in X$ such that

$$a^*(\psi, v) = F(v) \qquad \forall v \in X, \tag{1.11}$$

where X is a suitable function space and F is the cost functional (such as the drag), can be used to estimate what cells of a mesh contributed the most to the error on F.

The underlying assumption is that it is cheaper to perform n adaptation steps, each at the cost of solving $2n$ problems on meshes of increasing size – and cost – rather than solving only one expensive problem on a very fine mesh. As we will see below, this assumption is often justified in practice.

1.3 Software for numerical simulations

Software is one of the most important assets in the field of computational mathematics. Efficient, scalable software is essential to run huge simulations as the ones described in this work, to leverage the potential of the supercomputing clusters we have today. For the work I present in this thesis, I developed software based on FEniCS-HPC, a Finite Element framework written in C++ that automates the steps I outlined in Sections 1.1 and 1.2 [7, 15]. FEniCS-HPC is open source software licensed LGPL2 and offers functionalities to solve PDEs on massively parallel architectures, such as automated evaluation of variational forms which are specified in UFL – the Unified Form Language [2], a dialect of Python – in a high-level notation similar to mathematical notation; FEniCS-HPC also features automatic generation of variational forms for dual-based error control. FEniCS-HPC underwent thorough performance testing in the past and it repeatedly proved to scale with near-ideal performance on thousands of processes [12].

For fluid flow I contributed to the development of a solver called Unicorn [9], written on top of FEniCS-HPC, that we used for the results outlined in

Chapter 3. I also developed, for the purposes detailed in Chapter 4, a multi-physics simulation program that we internally call *the RFA code*, again on top of FEniCS-HPC and Unicorn, that was essential to the progress of our research.

1.3.1 Simulations in the cloud: primer

In recent years, supercomputing clusters, which used to be a niche asset of supercomputing centres, have become more common. Global companies like Google, Amazon and Microsoft possess enormous amounts of hardware, which they routinely use to perform their business tasks. However, these companies are now also selling computing time on their platforms to customers, together with bundled software for standard uses. This is the case of solutions like Amazon Web Services and Google Cloud Platform.

A natural question one might ask is if these resources and services can be used for scientific HPC too. If this were the case, then software like the one I write would be much easier to access by domain scientists – like physicians and doctors, but also chemists and astronomers – who are not well-enough versed in computer science to perform complex (and often unpredictable) installation procedures.

Apart from the accessibility point of view, cloud-based solutions can also be more economically viable. Traditionally, a research group or centre embarking on a journey involving HPC needed to carefully estimate its need for computing power, both in the short and in the long term, to strike a balance between cost and benefit acquiring enough hardware to run simulations in a reasonable time but not so much that it would remain unused. With cloud-based solutions, the user can effectively rent as much computational power as he or she needs for that single simulation and be charged for it; if the following time he or she needs more, it is always available. The concept of being able to flexibly allocate computing resources is sometimes referred to as *elastic computing*.

1.3.2 Simulations in the cloud: this thesis's contributions

I participated in a study of the feasibility of the use of this kind of solutions for scientific computing. Paper I reports on the results of this study, in which we show how to perform a simulation of a Magneto-Resonance Imaging model on Google Cloud Platform, and compare its performance with that of a traditional supercomputer. Incidentally, Paper I is also another nice example of a

field in which the potential of Finite Element simulations is not yet exploited to the fullest: in the field of Magneto-Resonance Imaging, statistical methods are more popular, despite being more computationally expensive for the same accuracy.

In a European project I contributed to, the *Mathematical Modelling, Simulation and Optimisation for Societal Challenges with Scientific Computing* (MSO4SC), we developed a web portal that lets users configure and run FEniCS-HPC simulations in the browser, without worrying about configuring, compiling or installing any software on their own. Tools like this portal, called *Marketplace* in the project, can help achieve the same goal of elastic computing. In this case, the Marketplace was not connected to Google or Amazon infrastructures, but to a traditional cluster in a supercomputing centre via a system that allocates resources on demand, but this makes no difference to the end-user.

Chapter 2

Adaptive Computational Fluid Dynamics with Finite Element Methods

In this chapter I will present the main computational tool in use in our research group, that we call the Direct Finite Element Simulation method, or DFS for short. I will explain what distinguishes DFS from other methods and how it can be much cheaper than what has been seen before. DFS is based on the contents of Sections 1.1 and 1.2, which are applied in the specific context of fluid dynamics, which I will first shortly introduce.

2.1 Fluid dynamics

Fluid dynamics is a very vast field. With basic ideas going back to Euler, fluid dynamics has since become a central part of mechanical, aerospace and civil engineering providing useful insight into how to build aeroplanes, cars and bridges but also heart valves and nuclear reactors. Despite being so important, some questions in fluid mechanics still do not have a satisfactory answer, one such example being turbulence, one of the last few open problems in classical mechanics that still eludes our comprehension. Part of the issue is that turbulence is too expensive to fully simulate in a computer because the features the flow develops are too fine for the resolution power of modern supercomputing clusters.

In this context I will only address incompressible, viscous fluid dynamics. The word *incompressible* refers to the fact that the fluid at hand does not show any compressibility effect, meaning that its density remains constant in time. This is a good approximation for all liquids and for gases under appropriate conditions, such as air at relatively low speeds and Mach number up to 0.3. The word *viscous* means that the particles of the fluid do not slip past each other frictionlessly, but instead experience friction from each other and can drag each other around. This is a reasonable assumption for almost all fluids that we encounter in engineering practice.

The most famous equations of fluid dynamics are the Navier-Stokes equations, a system of two equations encoding conservation of mass and momentum. For viscous, incompressible fluids, the former reads

$$\operatorname{div} \mathbf{u} = 0 \tag{2.1}$$

while the latter reads

$$\rho \partial_t \mathbf{u} - \mu \Delta \mathbf{u} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \tag{2.2}$$

where $\mathbf{u}$ is the fluid velocity, μ is the dynamic viscosity, p is the pressure, ρ is the density and $\mathbf{f}$ represents the external forces such as gravity.

The mass conservation equation (2.1) expresses the fact that the fluid is incompressible: a zero-divergence velocity field means, intuitively, that in any given point as much fluid is arriving as is leaving, implying that the total volume of the fluid remains constant. The momentum conservation equation (2.2) is a suitable reformulation of Newton's second law that forces $\mathbf{F}$ applied to matter change its momentum $\mathbf{Q}$ according to

$$\mathbf{F} = \dot{\mathbf{Q}}; \tag{2.3}$$

in the case of Equation (2.2), the momentum variation is $\dot{\mathbf{Q}} = \rho \dot{\mathbf{u}} = \rho \partial_t \mathbf{u} + \rho(\mathbf{u} \cdot \nabla)\mathbf{u}$ and the applied forces are $\mathbf{F} = \operatorname{div} \sigma(\mathbf{u}, p) = \mu \Delta \mathbf{u} - \nabla p + \mathbf{f}$, where $\sigma(\mathbf{u}, p) = 2\mu \operatorname{sym} \nabla \mathbf{u} - pI$ is the stress tensor and $\operatorname{sym} A = \frac{A + A^\top}{2}$ is the symmetric part of A; Equation (2.2) can then be compactly expressed as

$$\rho \dot{\mathbf{u}} = \operatorname{div} \sigma(\mathbf{u}, p), \tag{2.4}$$

beautifully expressing how derivatives capture the essence of our world.

For incompressible fluids, where density is constant, it is common to normalise the equations choosing a domain of unitary characteristic length and a flow of unitary characteristic speed, together with a unitary density. After

these changes of variables, but without changing notation for convenience, the equations read

$$\begin{cases} \partial_t \mathbf{u} - \nu \Delta \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p = \mathbf{f} \\ \operatorname{div} \mathbf{u} = 0 \end{cases} \tag{2.5}$$

where now $\nu = \frac{\mu}{\rho}$ is the kinematic viscosity and p and $\mathbf{f}$ have been implicitly divided by the density. The only parameter left, ν, is the ratio of the dynamic viscosity to the density and models the flow based on how strong diffusion effects are compared to inertial effects, and it is so important that its inverse is given the name of *Reynolds number* $Re = \nu^{-1}$. The Reynolds number, being the only nondimensional group left in Equation (2.5), determines the features of the resulting solution: in a flow with small Reynolds number, particles diffuse much faster than they drag each other around, resulting in smooth, laminar velocity profiles, whereas in a flow with high Reynolds number the opposite happens, and the velocity profile is turbulent and chaotic.

It should be noted that the real definition of the Reynolds number is

$$Re = \frac{LU}{\nu} \tag{2.6}$$

with L and U a characteristic length and speed of the system, such as, for example for the case of flow past a wing profile, the wing length and the far-field velocity. The definition I gave above keeps into account that Equation (2.5) was normalised such that $L = 1$ and $U = 1$.

2.2 Simulation of turbulent flows

DFS is a method, but it also a model. As such, its applicability and relevance are not universal but suit some scenarios better than others: one scenario in which DFS works particularly well is that of turbulent flows. This is not by chance: laminar (non turbulent) flows are significantly less expensive to compute because the flow field does not develop as many small features as their turbulent counterpart, making it is possible to fully resolve the flow field. We call this kind of simulation a Direct Numerical Simulation, or DNS. While DNSs give us all the information we want on the flowing fluid, it is not feasible to perform a DNS simulation of a turbulent flow. This is because we know from the theory [5] that the smallest scales of a flow field scale as $Re^{-\frac{3}{4}}$, so for the simulation of the flow past an aeroplane at a realistic Reynolds number of, say, ten millions, we would need an enormous number of degrees of freedom, far more than any supercomputer can currently elaborate.

The question with Computational Fluid Dynamics is then how to efficiently simulate flows at high Reynolds numbers, and DFS is an attempt to answer this question. Other notable attempts include Reynolds-Averaged Navier-Stokes equations (RANS), a family of models based on simulating time-averaged flows instead, replacing the nonlinearities in Equation (2.5) with heavy modelling intervention, and Large-Eddy Simulations, or LES, a family of models that behave like a DNS simulation up to a certain given scale, but do not resolve the smallest scales of the flow, which are replaced by some suitable turbulence model.

RANS simulations are very cheap but their heavy modelling intervention typically results in a number of parameters that should be tuned, or fitted, for most simulation scenarios with little support from a theoretical background. LESs, on the other hand, resemble reality much more closely but are more expensive than RANS simulations; in fact, LESs can take days to run on a realistic case [14], and in many cases they still require parameters to be found and specified.

2.3 DFS as a General Galerkin method for turbulent flows

The DFS method that we use for the simulation of turbulent flows consists of a Finite Element Method with linear elements for both the velocity and the pressure. Recalling Section 1.1, this means that in our framework both the velocity and the pressure are approximated, on a given mesh, by functions that are globally continuous over the domain and linear over each cell.

Since turbulent flows are *transport-dominated* flows, DFS relies on residual-based least squares stabilisation to avoid unphysical oscillations that would arise by the straightforward application of a FEM solver to Navier-Stokes equations. This is a well-known phenomenon in the literature of Finite Element simulation of transport phenomena. The innovation, here, is that the residual-based stabilisation is also used as a turbulence model. This acts in a similar way to how some LES models, called *implicit LES*, handle their subgrid scales.

2.3.1 Formulation of DFS

The DFS method is based on linear polynomial Finite Element in both time and space. Accordingly, the test functions will be continuous, piecewise linear

in space, but only piecewise constant in time. Let us subdivide the simulation time interval $[0,T]$ in a partition $0 = t_0 < t_1 < \ldots < t_N = T$ of time steps of length $\Delta t^n = t_n - t_{n-1}$ and let Ω be the domain of our simulation, on which we define a finite dimensional space $W \subset H^1(\Omega)$ of continuous, piecewise linear functions on the mesh $\mathcal{T}_h$ of mesh size $h = h(\mathbf{x})$; furthermore, let $W_{\mathbf{w}}$ be the subset of W of functions satisfying the Dirichlet boundary condition $\mathbf{u}_{\Gamma_D} = \mathbf{w}$. Then the DFS method for the Navier-Stokes equations with homogeneous Dirichlet boundary conditions can be formulated as: for every time step n find $(\mathbf{u}_h^n, p_h^n) = (\mathbf{u}_h(t^n), p_h(t^n)) \in [W_{\mathbf{0}}]^3 \times W$ such that

$$\left(\frac{\mathbf{u}_h^n - \mathbf{u}_h^{n-1}}{\Delta t^n} + \bar{\mathbf{u}}_h^n \cdot \nabla\, \bar{\mathbf{u}}_h^n, \mathbf{v}_h\right) + (2\nu\, \mathrm{sym}\, \nabla(\bar{\mathbf{u}}_h^n), \mathrm{sym}\, \nabla\, \mathbf{v}_h) - (p_h^n, \mathrm{div}\, \mathbf{v}_h) + (\mathrm{div}\, \bar{\mathbf{u}}_h^n, q_h) + SD_\delta^n(\bar{\mathbf{u}}_h^n, p_h^n; \mathbf{v}_h, q_h) = (\mathbf{f}, \mathbf{v}_h) \quad \forall (\mathbf{v}_h, q_h) \in [W_{\mathbf{0}}]^3 \times W. \quad (2.7)$$

In Equation (2.7), $\bar{\mathbf{u}}_h^n = \frac{\mathbf{u}_h^n + \mathbf{u}_h^{n-1}}{2}$, $\mathrm{sym}\, A = \frac{A + A^\top}{2}$ is the symmetric part of A, we added the aforementioned stabilisation term

$$SD_\delta^n(\bar{\mathbf{u}}_h^n, p_h^n; \mathbf{v}_h, q_h) = (\delta(\bar{\mathbf{u}}_h^n \cdot \nabla\, \bar{\mathbf{u}}_h^n + \nabla\, p_h^n - \mathbf{f}, \bar{\mathbf{u}}_h^n \cdot \nabla\, \mathbf{v}_h + \nabla\, q_h) + \delta\, (\mathrm{div}\, \bar{\mathbf{u}}_h^n, \mathrm{div}\, \mathbf{v}_h)\,, \quad (2.8)$$

$(\cdot,\cdot)$ represents the usual inner product, $\delta = \kappa h$ and κ is a positive constant of unitary size. For details on the choice of the stabilisation, see Paper II and Hoffman, Jansson, Jansson, et al. [8].

2.3.2 The infinite Reynolds number model

The way boundary conditions are modelled in DFS is one of the key contributors, alongside adaptivity, to the exceptionally low cost of a simulation, as compared to, for example, DNSs or LESs. In fluid dynamics it is traditionally accepted that the boundary layer plays an important role in the solution to the Navier-Stokes equations, and especially in the computation of aerodynamic forces such as lift and drag.

In high Reynolds number flows, the boundary layer originates from the fact that the velocity of the fluid in contact with the body surface is zero, but the velocity far from the boundary can be very high. The flow field then has a steep gradient going from zero to its far-field value very sharply. On top of it, the width w of the boundary layer goes to zero as the Reynolds

number approaches infinity, meaning that for very high Reynolds numbers the boundary layer is extremely thin, so much so that it requires a very thin mesh near the boundary to accurately be resolved. Much of the computational cost of traditional CFD methods lies in the cost of resolving the boundary layer.

In general, the boundary layer is, in fact, important, and the aerodynamic forces on a body, together with the flow features, change with the Reynolds number of the simulation. However, experiments have proven [1] that at very high Reynolds number these quantities become insensitive to the Reynolds number: while the lift on a wing, for example, changes greatly when the Reynolds number goes from 10 to 100 000, it does not change much for Reynolds numbers greater than 1×10^7. The fact that, for very high Reynolds numbers, aerodynamic forces tend to a limit motivates the introduction of an *infinite Reynolds number* model of the flow, meaning that we assume the flow, and in particular the aerodynamic forces, to be well approximated by the corresponding quantities computed with an infinite Reynolds number.

In practice, this means that the viscosity ν will be set to zero and that the boundary layer thickness will also be zero. The latter consequence is the most important one for efficiency: not having a boundary layer to resolve, a lot of computational cost can be saved that would otherwise be spent on resolving it.

The formulation of the boundary condition for our model then reads

$$\mathbf{u} \cdot \mathbf{n} = 0 \qquad \text{on } \Gamma_B \tag{2.9}$$

with $\Gamma_B \subset \partial\Omega$ the part of the boundary that covers the surface of the immersed body and $\mathbf{n}$ the unit vector normal to it. Similar conditions can be applied to lateral walls when simulating internal flow.

Adaptive error estimation

The last ingredient in DFS is adaptive error estimation. Once we have a solution to Equation (2.7), we need to decide where to refine the mesh, to prepare for the next iteration. To this end, we solve the dual problem of Equation (2.7), as outlined in Section 1.2. I should point out that the Navier-Stokes equations are a nonlinear system; the dual problem is uniquely defined, as in Equation (1.11), only for linear equations. In this case, one usually takes a linearisation of the initial equation, and uses the dual of that linearised version.

With the numerical solution to the dual problem, denoted by (φ_h, ϑ_h), consisting of the dual velocity and the dual pressure, one can compute an

estimate [8] of the error on the computation of a particular quantity of interest $M(\mathbf{u}, p)$, such as an aerodynamic force:

$$|M(\mathbf{u},p) - M(\mathbf{u_h}, p_h)| \leq \sum_n \left[\int_{I_n} \sum_{K \in \mathcal{T}_n} \|R_1(\mathbf{u}_h, p_h)_K \omega_1\| + |R_2(\mathbf{u}_h)_K \omega_2| + |SD_\delta(\mathbf{u}_h, p_h; \varphi_h, \vartheta_h)_K| \right]. \quad (2.10)$$

In Equation (2.10), $(\cdot,\cdot)_K$ denotes a scalar product restricted to the cell K, R_1 and R_2 are the strong residuals of the mass and momentum equations and

$$\begin{aligned} \omega_1 &= C_1 h_K \|\nabla \varphi_h\|_K \\ \omega_2 &= C_2 h_K \|\nabla \vartheta_h\|_K \end{aligned} \quad (2.11)$$

with C_1 and C_2 being some interpolation constants and $\|\cdot\|_K$ is a matrix norm on the cell K[1].

The DFS adaptive simulation methodology can thus be summarised as follows: given a mesh $\mathcal{T}_k$,

1. Solve the Navier-Stokes equations (2.7)
2. Solve the dual problem
3. Compute an error estimate with Equation (2.10)
4. If the estimate is greater than a given tolerance[2]
   (a) refine a fraction of the cells with highest error, generating $\mathcal{T}_{k+1}$
   (b) go back to point 1
5. Terminate the procedure

This is the other factor that contributes to DFS's great efficiency: with this simulation strategy, starting with a coarse initial mesh, and with the save in computational cost given by using slip boundary conditions to model boundary layers, DFS proved to be much cheaper than traditional methods, allowing to simulate time-dependent turbulent flows and not needing ad-hoc meshing, but rather being able to generate an optimal mesh as the iterative procedure progresses.

[1] The choice of the particular norm only affects the multiplicative constant.

[2] Or you reached a given maximum number of iterations.

Chapter 3

Verification test bed: high-lift configuration

This chapter describes the remarkable results that our software and simulation methodology achieved while participating in the 3rd AIAA High-Lift Prediction Workshop, held in Denver, USA on June 3rd-4th, 2017 and serves as an introduction to Paper 1. In the broader context of this thesis, the results described in this chapter serve as a validation of the methodology presented in Chapter 2.

The workshop's goal was to assess the status of current CFD simulation technologies, and in particular their ability to simulate a high-lift aeroplane configuration, which represents the configuration of an aeroplane at take-off and landing. This is all but a trivial problem: the main challenge today in CFD for aerodynamics is the ability to predict turbulent and separated flows [25]. The simulation of a full aircraft, as opposed to that of a more traditional wing profile, is even more challenging because the increased size of the computational domain increases the computational cost many times. Typical engineering studies only focus on wing profiles, the main devices for lift generation, and currently DNS simulations at realistic Reynolds numbers are only barely possible small for domains consisting of wing slices, and at an enormous computational cost; time-dependent LES and DNS simulations of a full aircraft are still a long way to come [24].

3.1 Problem description

The proposed benchmark is the simulation of flow past an aircraft model developed by the Japanese Aerospace Exploration Agency (JAXA) known as the JAXA Standard Model, or JSM, at a realistic Reynolds number of around 1.93×10^8. Figure 3.1 shows the wind tunnel setup that provided reference data for the benchmark. The main challenges in the simulation are the computation of drag and lift, with particular concern for the critical stall mechanism. Lift generation tends to increase with increased angles of attack of a wing. However, for a big enough angle of attack, the process abruptly interrupts and wings stop generating lift; this phenomenon is called stall and should be avoided as it makes the aeroplane unable to fly, possibly with dramatic consequences. For this reason, we had access to experimental results at a number of angle of attacks, from low values that are at no risk of stall to high values that exhibit stall. The challenge is, of course, to simulate stall at the correct angle of attack.

The workshop proposed two configurations of the JSM for simulation, one with a nacelle (the *pylon on* case) and one without a nacelle. Given the small difference in reference aerodynamic forces given by experiments in the two cases, a difference of about 2 %, we only focused on the *pylon on* test case, which is the most complex and realistic of the two.

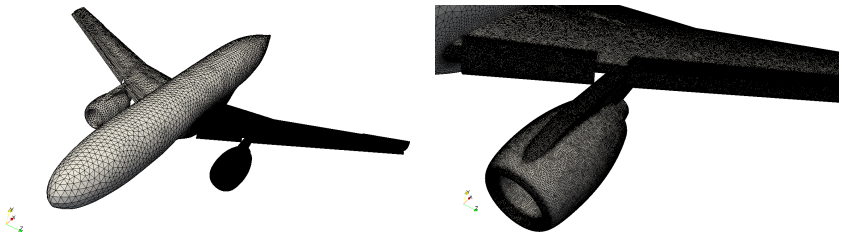
The organisers invited participants to test on two given meshes, a fine one to use as is and a coarser one to use with mesh adaptivity techniques. Since the method we develop uses mesh adaptivity, we opted for the second approach, but we still generated our own mesh from the provided CAD files because the given mesh was not coarse enough. It is worth pointing out at this point that our methodology can use a very coarse mesh as an initial mesh, the only requirement being that the geometric description of the surfaces, in this case of the aeroplane, be good enough to capture the geometric features. The reason for this is that adaptive refinement near the surface does not improve the surface description in the sense that new points are not projected on the CAD shape from which the mesh was generated, as no CAD information is kept in the mesh. This is a feature on which future development might focus as it would make meshing even easier and increase the overall automation of the whole process. Apart from this constraint, however, the volume mesh can be very coarse and the adaptive refinement will take care of adding mesh points where needed. In particular, no specialised meshing knowledge is needed, nor any a priori knowledge of the features of the solution.

3.2 Numerical tripping

The simulation of the stall cases was more challenging than the rest. This is because stall is, from a physical point of view, generated by flow separation, in particular at the wing tip and near the wing-body junction, that detaches the flow and negatively affects the generation of lift. The flow fields for the non-stalling configurations, which are well attached to the wing, are somewhat easier to reproduce; the transition to turbulence and consequent separation, on the other hand, were not. We noted, in our first attempts, that the flow was not separating as expected. Something similar happens with other numerical methods and the reason is that the inflow field, that is usually taken to be constant, is *too perfect* to accurately represent reality. In other words, no experiment in any wind tunnel will ever have a perfectly uniform and constant inflow, as is the case in numerical simulations, but rather an already chaotic inflow coming from air recirculation through the wind tunnel itself. The ideal inflow, combined with the numerical viscosity introduced by the term in Equation (2.8), makes it impossible for small perturbations to grow into the fully separated flow that we see in wind tunnel experiments.

Figure 3.1: A model of the JSM aeroplane in a wind tunnel.

In the literature, an effective workaround is to add a small white-noise volume source term to the equation that has the effect of simulating this random perturbation in the inflow field [23]. In an attempt to strike a balance between a term that goes unnoticed and one that dominates over other effects, we scaled the white noise by 5 % of the maximum of the pressure gradient. We called this method *numerical tripping* of the solution. Introducing numerical tripping has the effect of greatly reducing the initialisation time of a simulation, benefiting its overall computational cost.



(a) Surface mesh (b) Detail of wing mesh

Figure 3.2: The surface mesh we generated (left) and a detail of the mesh on the wing (right).

3.3 Results overview

I will not go into the detailed results analysis for the test case, which is reported in the attached Paper II, but rather give an overview of the results and discuss their significance.

We performed simulations for a number of angles of attack in the range $(4, 23)$ starting from an initial mesh of just around 2.5×10^6 vertexes. Unlike the experiment, which was performed on a half aeroplane profile, we created a mesh of a full profile to avoid modelling error introduced by a symmetry wall. Figure 3.2 shows the surface mesh on the aeroplane geometry that we created, with Figure 3.2b showing the finest bits where the geometry is more complex.

Our starting mesh is significantly coarser than the provided one and allows for very cheap simulations. At the end of our adaptive simulations, in which at each iteration we refined 5 % of the cells, the finest meshes we had counted only 5×10^6 to 1×10^7 vertexes; for comparison, the tetrahedral mesh provided by the workshop had around 2.1×10^7 vertexes, meaning that even on our final runs we spent between a half and a fourth of the cost of running on the provided mesh.

We observed mesh convergence to experimental data for the lift coefficient within 5 % for all the cases we ran, with a slightly bigger error of about 10 % for the drag coefficient. This is no surprise as the drag is notoriously more noisy and the typical drag signal oscillates noticeably, significantly more than the lift signal.

More importantly, for the configurations for which a stall is expected we were able to qualitatively simulate the stall mechanism, represented by a sharp

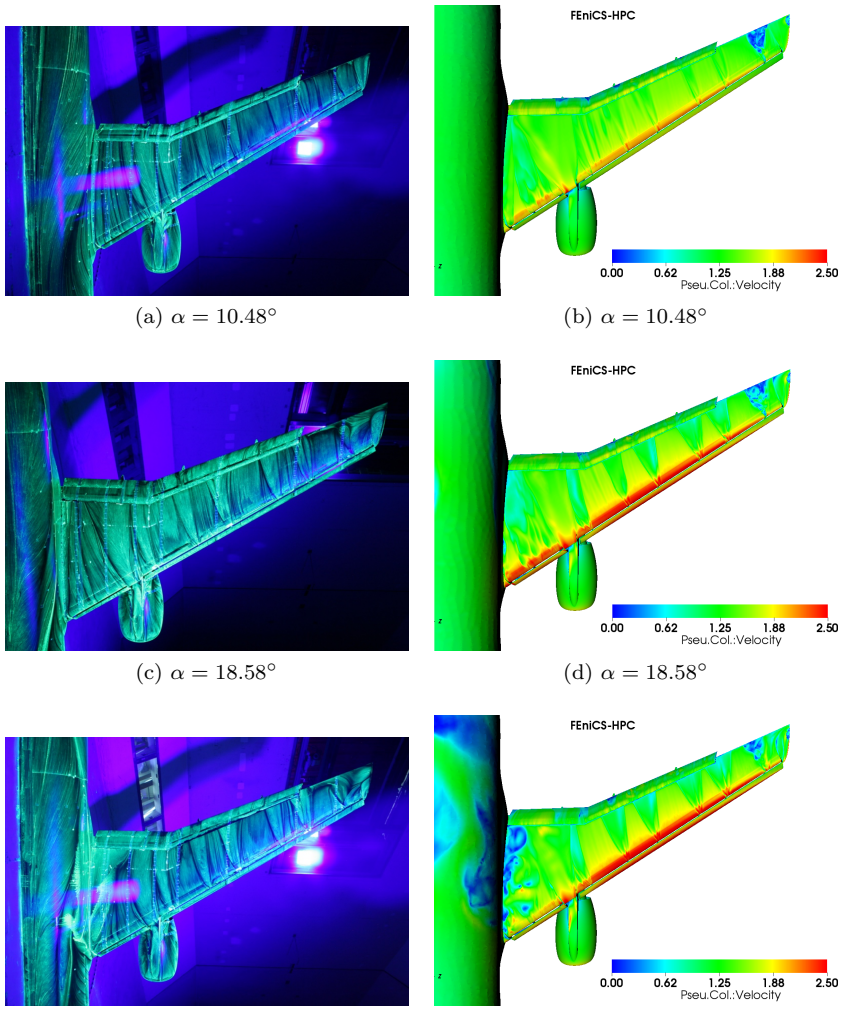
drop in the lift force, for an angle of attack which differs from the experimental one by about 1°.

Our simulations were the only ones to be time dependent, feature adaptivity and solve the full Navier-Stokes equations. Combining these features we were able to both predict aerodynamic forces to values within experimental ranges and reproduce the correct physical separation in the flow field. I would like to point out that nowhere in the code do we specify that we are interested in the separation, but we only ask for drag and lift. The correct prediction of separation, together with the pressure profiles and other local quantities are simply a by-product of the method, that we were still able to recover without explicitly asking for them.

As an example, consider the visualisation in Figure 3.3, where experimental data from surface oil flow visualisation is compared to our simulation results. What should be noticed in Figure 3.3 is that the simulation results closely match the profiles in the experimental setup, in particular the V-shaped structures originating from where the leading edge of the wing is connected to the flaps. In the first two rows the flow is attached as we are not at the stall angle yet, but the third row shows an important additional feature: separation at the wing-body junction, visually denoted by the blue, low velocity area on the left of Figure 3.3f.

To give another point of view on the same phenomenon, Figure 3.4 visualises the separation via the vorticity in the volume. It represents a snapshot of a fully-developed flow field, showing how the turbulent separation grows from the wing-body junction

Again, it is worth pointing out that the code was not tuned to resolve the separation, but only the aerodynamic forces; the rest is a by-product of the method that naturally resolves the correct physical mechanisms at play, in this case stall.



(a) $\alpha = 10.48°$ (b) $\alpha = 10.48°$

(c) $\alpha = 18.58°$ (d) $\alpha = 18.58°$

(e) $\alpha = 21.57°$ (f) $\alpha = 22.56°$

Figure 3.3: Comparison of surface oil flow visualisation from the wind tunnel (left) and velocity profiles (right) from our simulations.

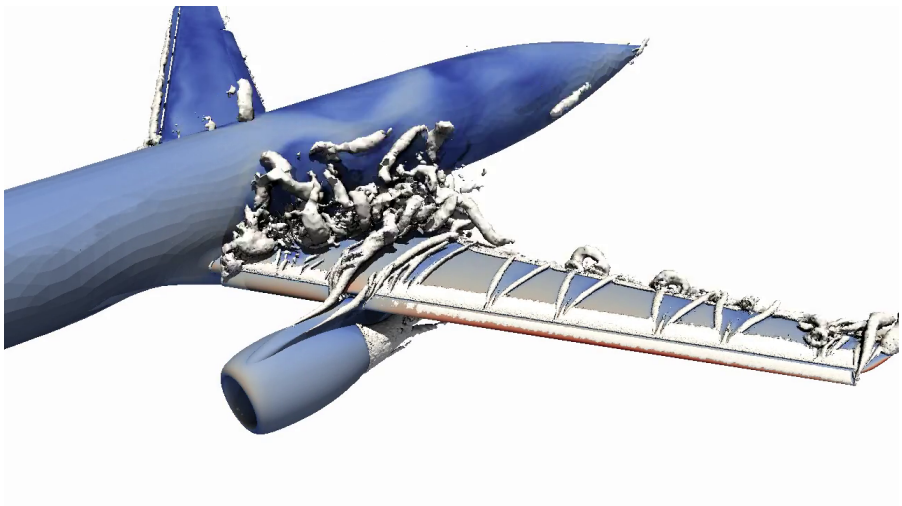


Figure 3.4: Visualisation of turbulent separation via vorticity computed with the Q-criterion. Here we superimpose a isosurface of the quantity $Q = \frac{\|\operatorname{skw} \nabla \mathbf{u}\|_2^2 - \|\operatorname{sym} \nabla \mathbf{u}\|_2^2}{2}$, with $\operatorname{skw} \nabla \mathbf{u}$ and $\operatorname{sym} \nabla \mathbf{u}$ the skew-symmetric and the symmetric part of the gradient of $\mathbf{u}$, respectively.

Chapter 4

Cardiac Radiofrequency Ablation

The rest of this thesis is devoted to a specific application to which I devoted a good share, if not the lion's share, of my time during my PhD: radiofrequency ablation.

4.1 Radiofrequency ablation

Radiofrequency ablation is a minimally invasive treatment in which a catheter is used to deliver heat to a target tissue with the goal of burning it [10]. Heat is delivered by an electric current at a radio frequency of about 500 kHz, whence the name of the procedure, and is used to permanently damage the tissue of the patient in a very localised manner. Radiofrequency ablation is routinely used to treat a number of conditions such as hyperopia, asthma, gastric reflux, hepatic tumour, apnoea and, last but not least, cardiac arrhythmia. The general idea of radiofrequency ablation is that causing controlled lesions in a tissue can be used to cure worse conditions, making it the lesser of two evils. Some cardiac arrhythmias, for example, are caused by electric impulses propagating in the heart to areas where they should not, eliciting muscular contractions that affect the correct pace of the heart. Burnt tissue, however, is known to be electrically insulating. Ablation can thus alter the electrical circuit in a patient's heart, and in particular we can cut unwanted branches, by generating lesions forming a barrier that electricity cannot cross.

Not all treatments are equally safe, and the treatment of cardiac disorders

is by far the most prone to risks and complications, even life-threatening. On average, around 5 % of the administrations of cardiac radiofrequency ablation result in unexpected complications, a much higher rate than the other use cases mentioned above.

The risks with cardiac radiofrequency ablation, from now on *RFA*, include the formation of cloths (or thrombi), when heat coagulates blood [3], and steam pop formation, arising when the temperature of the tissue is high enough that the water inside it evaporates [13]. Both scenarios are bad: thrombi have the potential to obstruct blood vessels if they accumulate in the circulatory system, whereas steam pops result in perforations of the cardiac wall. When the ablation is administered in the atria, which are characterised by thinner walls, the perforation can even be transmural, meaning it pierces the heart from side to side, putting the patient's life is in great danger. As a guideline, we consider that thrombi form in the blood when it reaches 80 °C, while steam pop occur in the tissue at a temperature of 100 °C; in contrast, tissue is considered irreversibly damaged at a temperature of 50 °C. The objects is then to efficiently generate large-enough lesions to block current propagation, by bringing the corresponding portions of cardiac tissue to a temperature of 50 °C, while making the procedure as safe as possible, by keeping the blood temperature below 80 °C and the tissue temperature below 100 °C.

RFA is administered to a sedated patient. The doctor performing the procedure pushes the catheter through a vein in the patient's body, entering from the hip, manually applying pressure. Several factors account for a less than ideal situation, most importantly the facts that the patient is not completely still – his heart is beating – and the doctor's pressure is not constant in time. All these contribute to the high variability in the quality of the results of the procedure. Although several different protocols are common, sometimes the lesions are too small, not blocking electricity properly and requiring a second administration, while sometimes too much power is delivered, resulting in complications. On top of it, during the administration of the procedure it is hard to assess the progress at the burning site, as inspecting the size of the lesion that is forming in the tissue is all but easy. It is clear that this scenario can benefit from the help of numerical simulations to better understand what happens at the patient during the procedure, and this is the mission we set off to accomplish. To be more specific, we want to develop a mathematical model and a simulation framework able to reproduce, in silico, an ablation procedure in order to be bale to provide clinicians and doctors useful information for the planning of RFA administration.

4.2 Lesion assessment

RFA is administered according to protocols. A *protocol* is a combination of applied contact force, power and time such as (20 g, 30 W, 10 s) meaning that the catheter is to be held applying a force of 20 grams, delivering 30 Watts of energy for 10 seconds.

Manufacturers of medical devices use models, to some extent, to advise medical doctors on what protocols to use for a session of radiofrequency ablation. These models are quite coarse-grained, often combining all three parameters in a protocol with some regression technique used a posteriori on experimental data from tests on animal tissue. Although many proved reasonably effective, these models suffer from the limitations caused by condensing too much information in a single number and lack physical connection to the real scenario, making them overly conservative. A mathematical model that draws from physical principles has the potential to do better.

4.3 Mathematical modelling

In this section I will go through the modelling intervention that we performed to build a so-called *digital twin* of the RFA procedure. All the novelties in this section were introduced in Paper III.

4.3.1 Governing equations

The RFA process is a multi-physics system involving fluid flow, current flux and thermal diffusion.

We used the incompressible Navier-Stokes equations to describe the fluid flow, which we already encountered in Equation (2.5) in Section 2.1 but that I report for convenience:

$$\begin{cases} \partial_t \mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} - \operatorname{div} \sigma(\mathbf{u}, p) = \mathbf{0} \\ \operatorname{div} \mathbf{u} = 0 \end{cases} \tag{4.1}$$

where $\mathbf{u}$ is the velocity, p is the pressure and $\sigma(\mathbf{u}, p) = 2\nu \operatorname{sym} \nabla \mathbf{u} - pI$ is the stress tensor.

The electrical potential can be described as the solution to a Poisson-type equation. This is a good approximation because the characteristic time in the evolution of radiofrequency currents is much shorter than that of the other phenomena considered in this simulation, so we can assume that the electrical

potential is quasi-static and avoid solving the whole Maxwell equations in favour of a simpler elliptic equation. We have

$$\operatorname{div}(\sigma(T)\,\nabla\,\Phi) = 0 \tag{4.2}$$

where Φ is the electrostatic potential and $\sigma(T)$ is the electrical conductivity, that we consider as a function of the temperature T.[1] This equation has to be solved with an additional constraint: during the RFA procedure, the machine used to provide power to the catheter does so trying to keep the dissipated power to a constant value P – the one that appears in the protocol. Recalling that, in electrostatics, the dissipated power is given by $P = \mathbf{E} \cdot \mathbf{J}$ where the electrostatic field is $\mathbf{E} = -\nabla\,\Phi$ and the electrostatic current is $\mathbf{J} = \sigma\mathbf{E}$, Equation (4.2) needs to be complemented with the constraint

$$\int_\Omega \sigma(T)\|\nabla\,\Phi\|^2 = P \tag{4.3}$$

Finally, for heat dissipation we used a modification of Penne's bioheat equation that describes tissue heating by both convection and power dissipation from radiofrequency currents. The final form, that we have been using in the model, reads

$$\rho c(T)\,(\partial_t T + \mathbf{u}\cdot\nabla\,T) - \operatorname{div}(k(T)\,\nabla\,T) = \sigma(T)\|\nabla\,\Phi\|^2 \tag{4.4}$$

featuring the density ρ, the specific heat c, the thermal conductivity k and the temperature T. Φ and σ are the same as in Equation (4.2) while $\mathbf{u}$ is taken from Equation (4.1).

4.3.2 Catheter geometry

For this model we have been using a geometry based on the in vitro experiment that we used as a validation case. The geometrical description, however, introduces a few improvements over the state of the art in these kinds of simulations.

First of all, the catheter geometry is quite complex in itself. The exterior of the catheter is an adiabatic body that insulates it thermally and electrically. At its tip, however, there is an electrode on which the RFA machine applies a potential difference to drive electric current. Inside the electrode tip there is a

[1] It is an unfortunate coincidence that both the stress tensor and the electrical conductivity are denoted by the letter σ in the literature of the respective fields, but in this work they have different arity so no confusion should arise.

glass piece, called *thermistor*, that has a high thermal capacity and is used to keep the electrode's temperature low, preventing it from overheating. Finally, at its core there is a channel that is used to inject coolant in the heart. To make things more complex, the channel has small outlets that pass through the thermistor and the electrode, in the count of six for the model used in the experiment. We report an overview of the catheter geometry in Figure 4.1.

Other works in the literature presented simulations of similar scenarios with a much simplified geometry: most in two spatial dimensions, with the ones in three dimensions having no description of the individual outlets of the inner channels, but instead a contiguous inlet surface going around the whole electrode. An important improvement in our model, in particular in the geometrical description, was to use meshes of very accurate geometries that resolved all the details of the catheter. We identify all the main parts of the catheter: the body, the electrode, the thermistor and the channels, with the electrode and the thermistor being part of the actual computational domain while the body and the inner channels are not but shape the boundary of the domain.

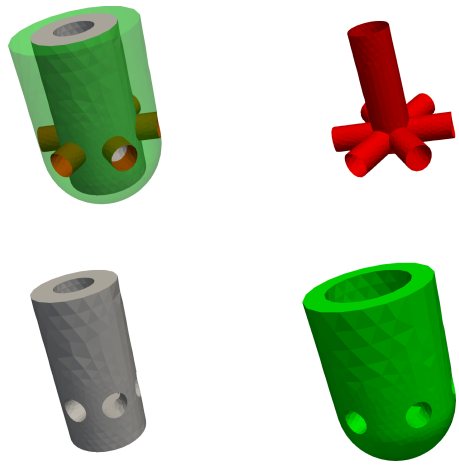


Figure 4.1: The components of the catheter tip: the inner channels in red, the thermistor in grey and the electrode in green.

4.3.3 External factors calibration

As far as the computational domain we used is concerned, it is composed by several additional parts. The catheter is inserted in a part of the domain occupied by blood, with its tip, the electrode, touching another part, occupied by the tissue. The tissue part is then resting on an additional part, which we labelled the *external factors board*, or *board* for short. In the experiment, the tissue lied on a methacrylate board and the dispersive electrode, necessary for the ablation, was not placed in direct contact with the tissue but rather under this additional board. The last part of the computational domain, then, corresponds to this additional object separating the tissue from the electrode.

I would like to point out that a similar situation happens in the administration of the real procedure where the dispersive electrode is, usually, placed on the skin of patient, usually on the back or on a shoulder, again rather far from the heart.

Figure 4.2 shows the complete computational domain , where we also reported its physical dimensions in meters for comparison.

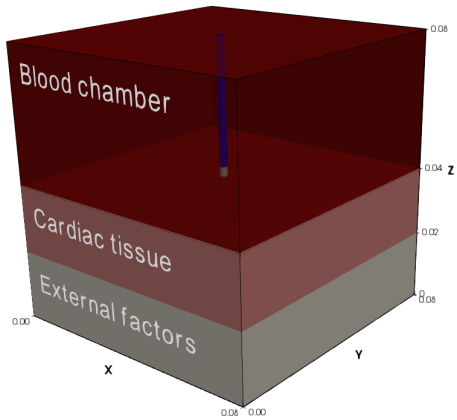


Figure 4.2: The computational domain used for our simulations, consisting of the catheter in the blood chamber, with its tip touching the cardiac tissue, which is resting on the external factors board. Dimensions in meters.

The presence of the external factors board also gives us the opportunity to calibrate the model to match the setup of the actual experiment, and this is where it gets its name from. The domain we used in our simulations is much smaller than the actual domain used for the experiments. This poses a problem as some of the electric power is dissipated outside the small volume we simulate, but it is difficult to keep track of it. Luckily, we know from the RFA machine how much resistance the whole setup offers to it, so we can tune the σ of the board in order to match the same value. This gives us a single parameter to tune the computational domain to simulate a much bigger volume; furthermore, the tuning is done only at the beginning of the simulation and the resulting value is maintained throughout the whole run. I wish to note that the constraint of matching the system's impedance is very important because it ensures that the correct amount of power is distributed to the various components of the domain – the blood, the tissue, etcetera – so much so that in a previous work [6] the same goal was achieved by changing the electrical impedance of the cardiac tissue. Our solution is more elegant and does not affect the physical properties of the parts of the system we are more interested into; we therefore consider it a significant improvement over the state of the art.

4.3.4 Elastic contact

The tip of the electrode is in direct contact with the tissue. During our investigation of the numerical simulation of RFA, it became clear how the geometry of this contact is of the utmost importance for an accurate simulation of the ablation procedure. The reason is that the part of the electrode surface that is in contact with the tissue will provide electric current directly to it, while the part that is in contact with the blood will dissipate energy into the blood. Electrons on the electrode, on which they can move freely as it is a good conductor, can flow either in the blood or in the tissue. We know from elementary theory of electromagnetism that electrons will flow into a medium in an amount directly proportional to the cross-sectional area of the medium and inversely proportional to its impedance; in other words, electrons will favour a medium that they can access through a large area and offers small resistance. The setting can be modelled as an electric circuit in which current can flow in two parallel branches, one containing a resistance representing the tissue, the other containing a resistance representing the blood and the tissue in a sequence. As detailed in the paper, following the analysis in Wittkampf and Nakagawa [26], with this description we deduce that the power dissipated in the tissue is proportional to

$$\alpha := \frac{A_{tissue}\sigma_{tissue}}{A_{blood}\sigma_{blood} + A_{tissue}\sigma_{tissue}} \tag{4.5}$$

It is clear that, to correctly reproduce a cardiac ablation, it is essential to know how much of the electrode is in contact with the tissue and how much with the blood; in other words, we need to know the geometry of the electrode-tissue contact, or our simulation, even with all the correct values for the various physical parameters, will behave differently from reality. It is important to highlight that the electrode-tissue contact surface depends on the applied contact force which is specified in the RFA protocol.

We introduced a model of elastic contact to take this aspect into account. We modelled the configuration as an indentation problem of a rigid sphere – the catheter tip – into a half-space – the cardiac tissue. We chose to use an elastic material description for the tissue as this gives us the opportunity to use realistic values for the Young modulus and the Poisson ratio that are known in the literature, and also because it is a reasonable choice. We borrowed knowledge from contact mechanics and described our indentation problem via an *axisymmetric Boussinesq formulation*, which gives us the profile of the tissue surface for a given applied force. Prior to the use of contact mechanics to compute a contact profile, numerical models of RFA simply considered a sharp

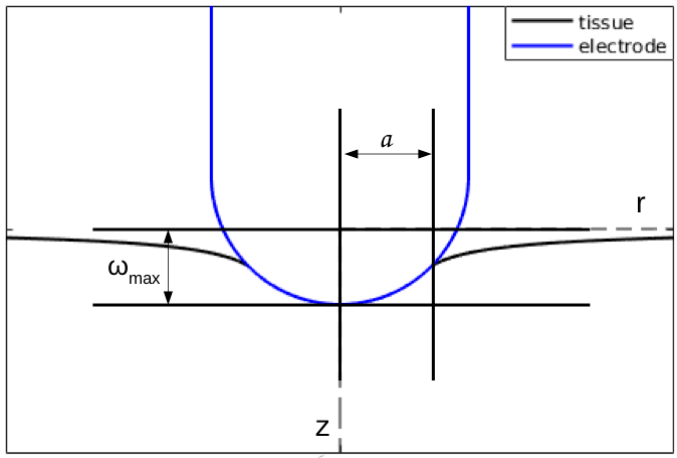
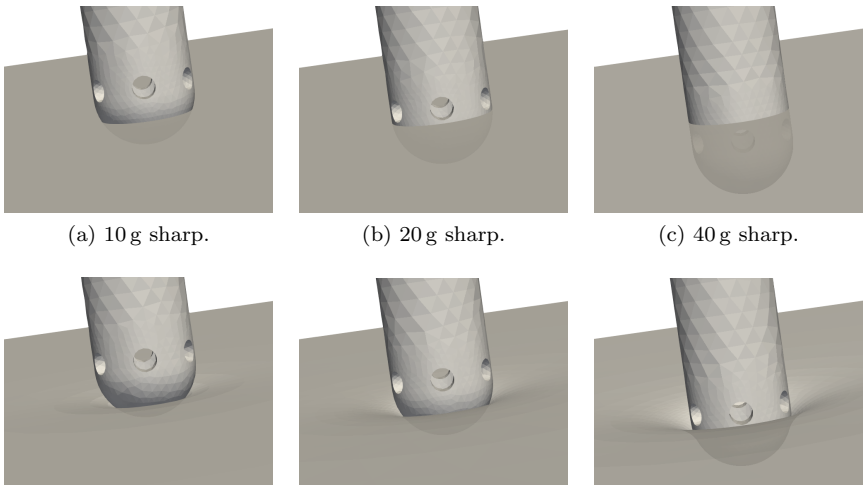


Figure 4.3: Sketch of the elastic contact profile: the tissue surface deforms to follow the indenting sphere.

insertion of the catheter into the tissue, meaning that the penetration depth of the catheter into the tissue was adjusted based on the exerted force, but the tissue surface was left unchanged (flat). This led to an intrinsic overestimation of the electrode-tissue surface contact area that jeopardised the physics of the whole system. A visual comparison of the two methods will make the difference clear: Figure 4.4 presents such comparison. It immediately strikes the eye how one of the two approaches, the one represented in the bottom row of pictures, is more realistic. If the difference is not dramatic in the 10 g case, it is in the other two, especially in the 40 g case where the displacement is such that the irrigation holes end up under the tissue surface, a clearly unphysical scenario. The elastic deformation model that we introduced gives a much more realistic approximation of the electrode-tissue contact, resulting in a correct distribution of the electric current between the blood and the tissue.

I want to stress that the misbehaviour caused by a computational mesh with an incorrect electrode-tissue contact is not caused by a parameter that can be tuned in the code, because it is a property of the shape of the solution to Equation (4.2). Although one can still, despite an incorrect geometry, manage to deliver the correct amount of power to the tissue, this will affect the results of the simulation in the rest of the domain, for example in the blood, and ultimately deteriorate the physical accuracy of the simulation in a



(a) 10 g sharp. (b) 20 g sharp. (c) 40 g sharp.

(d) 10 g elastic. (e) 20 g elastic. (f) 40 g elastic.

Figure 4.4: Comparison of tissue-catheter contacts for sharp (top row) and elastic (bottom row) contact deformation models for three common values of pressure.

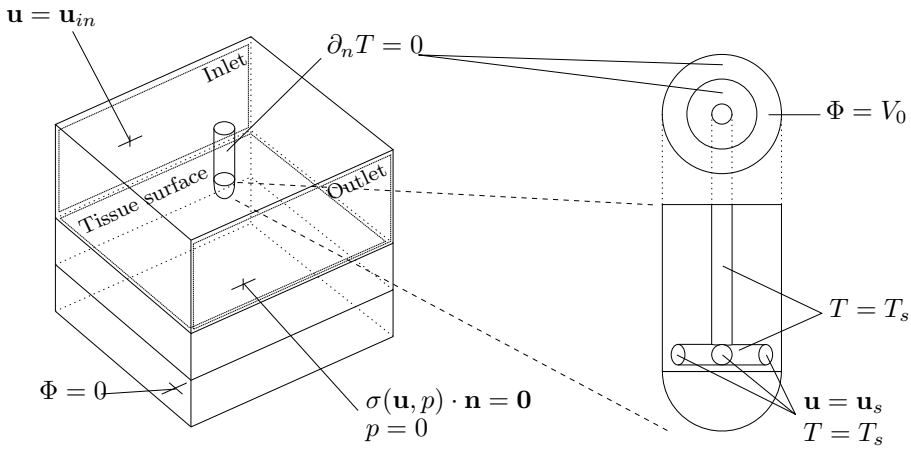


Figure 4.5: Schematic representation of our geometry and boundary conditions that differ from no-slip ($\mathbf{u} = \mathbf{0}$), zero current flux ($\sigma(T) \nabla \Phi \cdot \mathbf{n} = 0$) and body temperature ($T = T_b$). On the right, a detail of the catheter's tip in front and top view.

way that cannot be corrected.

Adding our elastic contact model was a crucial step toward a successful model and was very effective, while it also helped us better understand the physics of the radiofrequency ablation problem.

4.3.5 Model summary

Our model is ruled by Equations (4.1), (4.2) and (4.4) on a mesh generated according to the discussion in the preceding subsections. These equations are equipped with a variety of boundary conditions: Figure 4.5 shows a schematic representation of the domain, a detail of the catheter tip and all boundary conditions that differ from no-slip ($\mathbf{u} = \mathbf{0}$), zero current flux ($\sigma(T) \nabla \Phi \cdot \mathbf{n} = 0$) and body temperature ($T = T_b$). In this figure, $\mathbf{u}_{in}$ is the blood's inflow velocity, V_0 is the applied potential, $\mathbf{u}_s$ is the saline's inflow velocity and T_s is its temperature.

The choice of zero current flux on the boundaries of the domain is motivated by the need to make all the current flow to the dispersive electrode, which is what happens in reality. However, the real domain is much bigger than the computational one from which some current should leave, so an even

better approach would be to try and estimate the current that leaves our small domain and impose that flux on the boundary instead.

4.4 Numerical discretisation

The equations presented in Subsection 4.3.1 are too difficult to be solved in closed form, which means we cannot find an explicit expression for the velocity, pressure, potential and temperature solving them. This is due to the fact that the equations are both complex in their nature, being coupled and nonlinear, and defined on a complex geometry, the one described in Subsection 4.3.2. Luckily, the Finite Element Method, introduced in Section 1.1, can help us find an approximate solution to these equations despite their complexity.

Here we see the power of a general technique that is agnostic of the particular shape of the equation it is applied to or the domain the equation is defined on. For example, the only requirement the equation's domain has to satisfy is to be *sufficiently regular*, which is a technical way of saying that the technique fails on shapes that have pathological characteristics such as infinitely small edges or unfathomably irregular borders. Without going into the details of this issue, I want to point out that most, if not all, objects of the real world satisfy this criterion, including the ablation domain of our simulations. Once we have this, we can proceed as in Section 1.1 and derive a variational formulation of Equations (4.1), (4.2) and (4.4), similarly to how we derived Equation (1.6) from Equation (1.1), and finally a Finite Element approximation, analogous to Equation (1.8)

The Finite Element formulation of Equation (4.1) is basically the same as Equation (2.7), while the Finite Element formulation of Equation (4.2) is almost identical to Equation (1.8), with only the addition of the factor $\sigma(T)$ and no right-hand side. The weak formulation of Equation (4.4) can be derived similarly, and I report it here for completeness: for every time step n find $T_h^n = T_h(t^n) \in V_h$ such that

$$\int_\Omega \rho c(T_h^{n+1}) \left(\frac{T_h^{n+1} - T_h^n}{\Delta t} + \mathbf{u}_h^{n+1} \cdot \nabla T_h^{n+1} \right) v + k(T_h^{n+1}) \nabla T_h^{n+1} \nabla v = \int_\Omega \sigma(T_h^{n+1}) \left\| \nabla \Phi_h^{n+1} \right\|^2 v \qquad \forall v \in V_h \quad (4.6)$$

where, of course, Δt is the time step size, $\mathbf{u}_h^{n+1}$ and Φ_h^{n+1} are the discrete velocity and potential at the $n+1$-th time step and V_h is a suitable function space that incorporates the boundary conditions.

I would like to point out that, although the equations presented in Subsection 4.3.1 describe biological phenomena, little to no specialised knowledge of biology is required to be able to discretise and solve them. For this reason, abstract techniques in Mathematics and Computational Sciences can be seamlessly applied to problems arising from aerospace and biomedical engineering alike, and to a variety of other interesting fields.

4.5 Experimental setup and data

A good model is not very incisive if it does not match experimental data, which in the case of clinical procedures is often scarce. In the case of RFA, luckily, we had access to an experimental facility where we were able to perform a small number of test ablations in order to collect some data. These tests were not performed on humans, but rather on porcine cardiac tissue; this only affects the values of some physical parameters that we input to the code, but not the general idea and validity of our model. The experimental setup consists of the catheter and the cardiac tissue, which is resting on a plastic slab. This setup is submerged into a blood bath, which in turn is contained in a thermal reservoir. A blood pump circulates the blood to simulate the flow of blood typical of a living patient, which is the main means of thermal dissipation. The setup is pictured in Figure 4.6.

With this setup, clinicians perform a radiofrequency ablation according to a given protocol on the tissue. They later cut the tissue open where the ablation was performed and wash it with a special solution that, reacting with the burnt tissue, renders it white. This makes it possible to have a visual feedback of the lesion that the procedure generated. Figure 4.7a shows one such lesion, from the very same set of experiments. For comparison, Figure 4.7b shows a similar lesion as a result of a simulation using the model we developed. In Figure 4.7 we highlighted the definitions of some quantities of interest that describe a lesion's shape, namely its width (W), its depth (D) and the depth of its maximum width (DW). These quantities are very meaningful as they can be used to plan an administration of the procedure, and they are easy to compute both in vitro and in silico. We validated our code and model against this data; all the details are in the attached paper but a brief overview of the results is in Section 4.6.

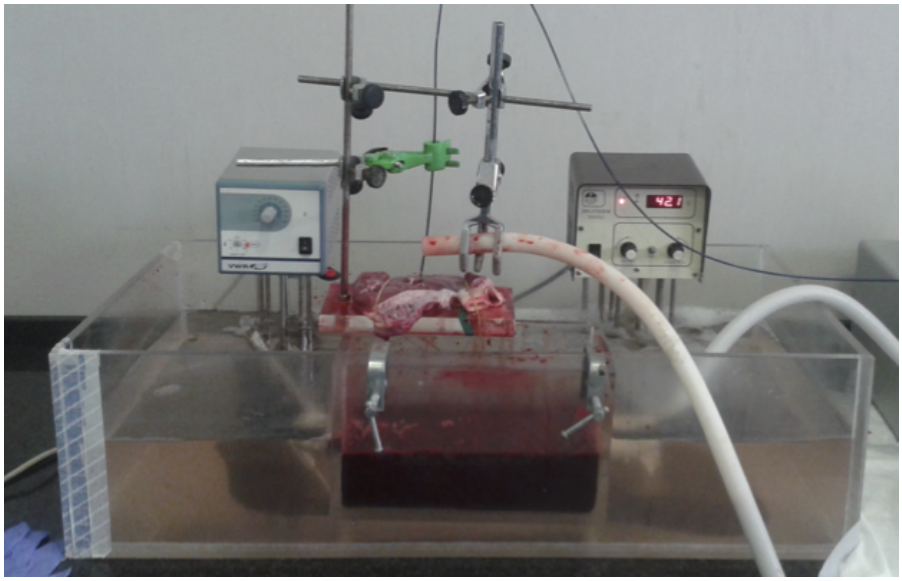
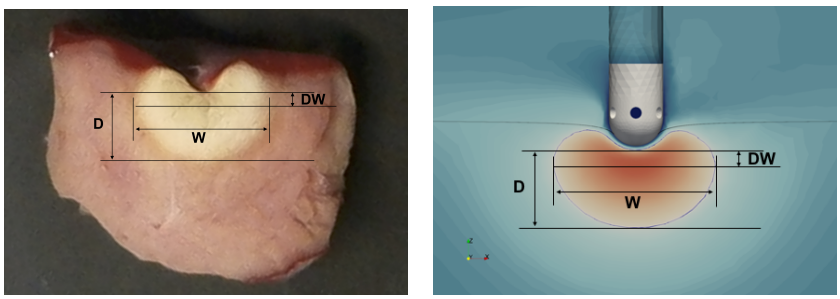
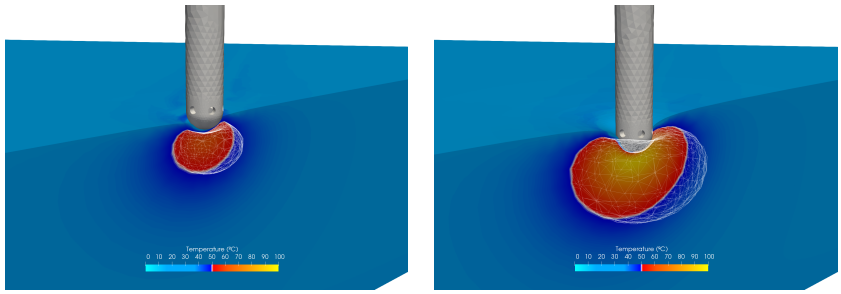


Figure 4.6: The experimental setup that we aim to reproduce with our model: the tissue and the electrode are submerged in a blood bath where blood flows thanks to a pump.



(a) Experimental lesion. (b) Simulated lesion.

Figure 4.7: A lesion caused to real tissue in an experiment and a simulated lesion obtained with our code. On both figures we defined the quantities of interest that we used to validate our model against the available experimental data.



(a) 10 g (b) 40 g

Figure 4.8: Simulated lesions for two protocols at 10 g (left) and 40 g (right). The view shows a cut through the cardiac tissue and an isosurface at 50 °C that represents the lesion.

4.6 Results overview

This section contains sample results from simulations of the model introduced in Section 4.3. I do not want to delve into the details of data validation in favour of a more qualitative presentation: a quantitative discussion is deferred to the attached Paper III; here, it is enough to anticipate that we were able to validate the experimental data to a satisfactory level of accuracy.

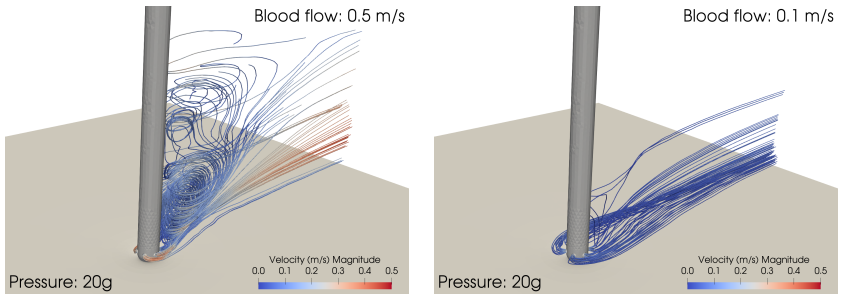
Instead, I would like to show a typical visualisation of a simulation result. Figure 4.8 shows two lesions simulated with protocols that differ in the applied pressure. The lesions are identified by the white isosurface, while the simulation also gives an estimate of the temperature reached inside the lesion, which is critical as we discussed in Section 4.1. The results are consistent with our expectations: the lesion in Figure 4.8b is much bigger than that in Figure 4.8a, a consequence of the fact that the electrode-tissue contact area is bigger. Datasets like those behind Figure 4.8 contain much more information than the various indexes used to estimate the lesion size that I mentioned in Section 4.2, making them much more useful for clinical decision-making. With detailed information on the size of the lesion it is possible to optimally plan the administration of the procedure to reduce the risk for the patient and improve the result.

4.7 Cylindrical catheters, high and low flows, high-power ablation and beyond

Availability of experimental data is often low in medical applications. Even when a suitable experimental facility is present, as was the case in this project, the amount of useful data that can be generated is limited, making it harder to test new protocols and techniques. A reliable model makes the task easier as it allows to experiment at will in the digital twin with confidence that the behaviour is representative of the actual phenomenon. For this reason, the model we built can be, and has been, used to test the effects of variations in the geometry, in the parameters, in the protocol, etcetera. In this section I would like to point out possible present and future research directions that we can follow using the technology we developed.

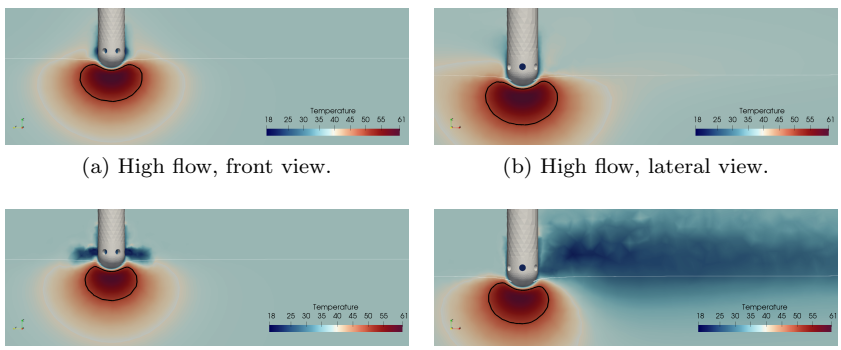
- **Cylindrical catheters.** The catheters we simulated when testing the model all have a spherical tip, in order to match the one used in the experiments we tried to reproduce. However, cylindrical tips are also common and one might ask if one of is better than the other. Cylindrical tips are flat and their electrode-tissue contact area changes much less with a perturbation in the applied pressure than the one of spherical catheters. Knowing how important the electrode-tissue surface area is, one might argue that cylindrical tips offer more stability in the results being less sensitive to variations in the applied pressure, that are systemic and difficult to eliminate completely. Since simulating a cylindrical tip is very easy with out model, as we only need to provide the code with a corresponding mesh, we can easily investigate this question; as a matter of fact, we are currently doing it.

- **High flow and low flow.** The blood flow in the heart during the ablation is the main means of thermal diffusion that, alongside the coolant injected by the catheter, helps to control the temperature rise. In our simulation we mainly used a fixed value for the inlet of the blood flow, a reasonable value that matched the experimental setup. In the human heart, however, the blood velocity varies noticeably from atria to ventricles, and in sedated patients it is likely smaller than in a regular heart. Figure 4.9 shows how the convection of the saline coolant is affected by the blood flow we routinely used (that we call *high flow*, Figure 4.9a) and by a weaker one (which we refer to as *low flow*, Figure 4.9b). In Figure 4.10 we show a comparison of lesions from simulations with high and low profiles, in front and lateral views. Interestingly, a low blood



(a) High flow (b) Low flow

Figure 4.9: Convection of the saline coolant with a high flow (left) and a low flow (right) profiles.



(a) High flow, front view. (b) High flow, lateral view.

(c) Low flow, front view. (d) Low flow, lateral view.

Figure 4.10: Lesion comparison with high and low flow profiles.

flow makes the coolant more effective as the lesion is overall smaller (Figure 4.10a versus Figure 4.10c) and asymmetric, as the cooling becomes more effective in the wake of the cylinder where the coolant stagnates a little longer (Figure 4.10b versus Figure 4.10d).

- **High-power ablation.** Recently, clinicians showed a lot of interest for *high-power* ablation protocols, meaning protocols that prescribe ablations dissipating powers in the range 70 W-90 W, significantly higher than what is common at the time of writing, but for shorter periods of time. The benefits would include reduced discomfort for the patient, from a shorter procedure, and in general a shorter time window in which complications can arise. We have been investigating high-power ablation protocols using our model and we collected the results in Paper VI. Again, from a practical point of view this only meant changing a number in our configuration and running several simulations.

- **Human versus animal tissue.** The experiments we reproduced were performed on porcine cardiac tissue. However, human cardiac tissue has different mechanical properties and, a priori, it is not clear if the results will be the same in terms of lesion sizes and maximum temperatures. Our model lets us verify this assumption very easily by simply setting different material properties for the tissue, resulting in a different electrode-tissue contact deformation, a parameter to which the simulation results are very sensitive, as we discussed. We collected results on this issue in a scientific contribution that is attached in the second part of this thesis.

The above list is not exhaustive and hopes to express the relevance of the model that we implemented and the number of research directions it allows us to investigate.

Chapter 5

Discussion and conclusions

In this thesis I presented the work I did on the development of software and methods in high-performance computing, specifically in the area of Finite Element simulations and applications.

In this framework, Paper I describes how such a technology can be delivered to domain scientists quickly and seamlessly, exploiting modern cloud-computing environments, and in particular containerisation technologies. Paper II verifies the particular method and software that we developed on an extremely challenging test case using a massively parallel computing platform. Paper III describes the development and implementation of a new model, and the necessary software to simulate it, to study a complex, multi-physics problem in biomedicine: Cardiac Radiofrequency Ablation. Papers IV, V and VI report on scientific results that followed from the model and tools developed in Paper III. Although they can be seen as different applications of the same methodology, their very nature represents the fact that the aim stated at the beginning of this work was, in fact, reached. This is true because the availability of an effective model, alongside its computer implementation, allows us – and potentially other researchers too – to mainly focus on the problem (its physics, its true nature) and to quickly find the answers to critical research questions such as *How relevant is the fact that protocols are tested on animals and administered to humans?* (Paper IV), *What happens if I use a cylindrical tip instead of a spherical one?* or *Can I improve on existing protocols by delivering more power for shorter time intervals?* (Paper VI). Paper VI deserves a special mention as it was submitted to a *clinical* journal. Although not yet accepted at the time of this writing, the publication of this paper would mark a milestone in the process of bridging the gap between theory and practice,

bringing our developments to the direct attention of medical doctors who are the their intended final users.

Bibliography

[1] H. I. Abbott, A. E. Von Doenhoff, and L. S. Stivers. *Summary of airfoil data*. Tech. rep. 1945, p. 399.

[2] M. S. Alnæs et al. "Unified Form Language: A domain-specific language for weak formulations of partial differential equations". In: *ACM Transactions on Mathematical Software* 40.2 (Nov. 2012), pp. 1–37.

[3] J. M. Demolin et al. "Soft Thrombus Formation in Radiofrequency Catheter Ablation". In: *Pacing and Clinical Electrophysiology* 25.8 (2002), pp. 1219–1222.

[4] L. Evans. *Partial Differential Equations*. Vol. 19. Graduate Studies in Mathematics. Providence, Rhode Island: American Mathematical Society, Mar. 2010.

[5] U. Frisch and R. J. Donnelly. "Turbulence: The Legacy of A. N. Kolmogorov". In: *Physics Today* 49.11 (Nov. 1996), pp. 82–84.

[6] A. González-Suárez et al. "Computational modeling of open-irrigated electrodes for radiofrequency cardiac ablation including blood motion-saline flow interaction". In: *PloS one* 11.3 (2016), e0150356.

[7] J. Hoffman, J. Jansson, and N. Jansson. "FEniCS-HPC: Automated Predictive High-Performance Finite Element Computing with Applications in Aerodynamics". In: *Parallel Processing and Applied Mathematics*. Springer International Publishing, 2016. Chap. 11th Inter, pp. 356–365.

[8] J. Hoffman et al. "Computability and adaptivity in CFD". In: *Encyclopedia of Computational Mechanics*. Vol. to appear. Chichester, UK: John Wiley & Sons, Ltd, Oct. 2016, pp. 1–40.

[9] J. Hoffman et al. “Unicorn : Parallel adaptive finite element simulation of turbulent flow and fluid-structure interaction for deforming domains and complex geometry”. In: *Computers&fluids* (2012).

[10] S. Huang and J. M. Miller. *Catheter Ablation of Cardiac Arrhythmias*. Elsevier, 2011, p. 803.

[11] J. Jansson et al. “Time-resolved adaptive direct fem simulation of high-lift aircraft configurations”. In: *Numerical Simulation of the Aerodynamics of High-Lift Configurations*. Cham: Springer International Publishing, 2018, pp. 67–92.

[12] N. Jansson, J. Hoffman, and J. Jansson. “Framework for Massively Parallel Adaptive Finite Element Computational Fluid Dynamics on Tetrahedral Meshes”. In: *SIAM Journal on Scientific Computing* 34.1 (Jan. 2012), pp. C24–C41.

[13] J. S. Koruth et al. “Occurrence of steam pops during irrigated RF ablation: Novel insights from Microwave Radiometry”. In: *Journal of Cardiovascular Electrophysiology* 24.11 (2013), pp. 1271–1277.

[14] J. Larsson and Q. Wang. “The prospect of using large eddy and detached eddy simulations in engineering design, and the research required to get there”. In: *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences* 372.2022 (Aug. 2014), p. 20130329.

[15] A. Logg and G. N. Wells. *Automated Solution of Differential Equations by the Finite Element Method*. Ed. by A. Logg, K.-A. Mardal, and G. Wells. Vol. 84. Lecture Notes in Computational Science and Engineering. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, p. 724.

[16] V.-d. Nguyen et al. “Portable simulation framework for diffusion MRI”. In: *Journal of Magnetic Resonance* 309 (Dec. 2019), p. 106611.

[17] A. Petras et al. “A computational model of open-irrigated radiofrequency catheter ablation accounting for mechanical properties of the cardiac tissue”. In: *International Journal for Numerical Methods in Biomedical Engineering* 35.11 (Nov. 2019).

[18] A. Petras et al. “Effect of Tissue Elasticity in Cardiac Radiofrequency Catheter Ablation Models”. In: *2018 Computing in Cardiology Conference (CinC)*. Vol. 2018-Septe. IEEE, Dec. 2018, pp. 2018–2021.

[19] A. Petras et al. “Tissue Drives Lesion: Computational Evidence of Interspecies Variability in Cardiac Radiofrequency Ablation”. In: *Functional Imaging and Modeling of the Heart*. 2019, pp. 139–146.

[20] A. Quarteroni. *Numerical Models for Differential Problems*. Milano: Springer Milan, 2009, p. 618.

[21] S. S. Rao and S. N. Atluri. "The Finite Element Method in Engineering". In: *Journal of Applied Mechanics* 50.4a (Dec. 1983), pp. 914–914.

[22] S. Salsa. *Partial Differential Equations in Action*. 2008, p. 568.

[23] P. Schlatter and R. Örlü. "Turbulent boundary layers at moderate Reynolds numbers: inflow length and tripping effects". In: *Journal of Fluid Mechanics* 710 (Nov. 2012), pp. 5–34.

[24] P. Spalart. "Strategies for turbulence modelling and simulations". In: *Engineering Turbulence Modelling and Experiments 4*. Elsevier, 1999, pp. 3–17.

[25] F. D. Witherden and A. Jameson. "Future Directions in Computational Fluid Dynamics". In: *23rd AIAA Computational Fluid Dynamics Conference*. Reston, Virginia: American Institute of Aeronautics and Astronautics, June 2017.

[26] F. H. Wittkampf and H. Nakagawa. "RF Catheter Ablation: Lessons on Lesions". In: *Pacing and Clinical Electrophysiology* 29.11 (Nov. 2006), pp. 1285–1297.

Part II

Papers

Journal of Magnetic Resonance 309 (2019) 106611

Contents lists available at ScienceDirect

Journal of Magnetic Resonance

journal homepage: www.elsevier.com/locate/jmr

Portable simulation framework for diffusion MRI

Van-Dang Nguyen [a,*], Massimiliano Leoni [a], Tamara Dancheva [b,a], Johan Jansson [a], Johan Hoffman [a], Demian Wassermann [c], Jing-Rebecca Li [d]

[a] Division of Computational Science and Technology, KTH Royal Institute of Technology, Sweden
[b] Basque Center for Applied Mathematics (BCAM), Bilbao, Spain
[c] Parietal, INRIA, Paris, France
[d] INRIA Saclay-Equipe DEFI, CMAP, Ecole Polytechnique Route de Saclay, 91128 Palaiseau Cedex, France

ARTICLE INFO

Article history:
Received 1 August 2019
Revised 8 September 2019
Accepted 19 September 2019
Available online 23 September 2019

Keywords:
Cloud computing
Diffusion MRI
Bloch-Torrey equation
Interface conditions
Pseudo-periodic boundary conditions
FEniCS

ABSTRACT

The numerical simulation of the diffusion MRI signal arising from complex tissue micro-structures is helpful for understanding and interpreting imaging data as well as for designing and optimizing MRI sequences. The discretization of the Bloch-Torrey equation by finite elements is a more recently developed approach for this purpose, in contrast to random walk simulations, which has a longer history. While finite element discretization is more difficult to implement than random walk simulations, the approach benefits from a long history of theoretical and numerical developments by the mathematical and engineering communities. In particular, software packages for the automated solutions of partial differential equations using finite element discretization, such as FEniCS, are undergoing active support and development. However, because diffusion MRI simulation is a relatively new application area, there is still a gap between the simulation needs of the MRI community and the available tools provided by finite element software packages. In this paper, we address two potential difficulties in using FEniCS for diffusion MRI simulation. First, we simplified software installation by the use of FEniCS containers that are completely portable across multiple platforms. Second, we provide a portable simulation framework based on Python and whose code is open source. This simulation framework can be seamlessly integrated with cloud computing resources such as Google Colaboratory notebooks working on a web browser or with Google Cloud Platform with MPI parallelization. We show examples illustrating the accuracy, the computational times, and parallel computing capabilities. The framework contributes to reproducible science and open-source software in computational diffusion MRI with the hope that it will help to speed up method developments and stimulate research collaborations.

1. Introduction

The numerical simulation of the diffusion MRI signal arising from complex tissue micro-structures is helpful for understanding and interpreting imaging data as well as for designing and optimizing MRI sequences. It can be classified into two main groups. The first group is referred to as Monte-Carlo simulations in the literature and previous works include [1–5]. Software packages include the UCL Camino Diffusion MRI Toolkit [6], which has been widely used in the field. The second group of simulations relies on solving the Bloch-Torrey PDE in a geometrical domain, either using finite difference methods (FDM) [7–10], typically on a Cartesian grid, or finite element methods (FEM), typically on a tetrahedral grid. Previous works on FEM include [11] for the short gradient pulse limit of some simple geometries, [12] for the multi-compartment Bloch-Torrey equation with general gradient pulses, and [13] with the flow and relaxation terms added. In [14], a simplified 1D manifold Bloch-Torrey equation was solved to study the diffusion MRI signal from neuronal dendrite trees. FEM in a high-performance computing framework was proposed in [15,16] for diffusion MRI simulations on supercomputers. An efficient simulation method for thin media was proposed in [17]. A comparison of the Monte-Carlo approach with the FEM approach for the short pulse limit was performed in [11], where FEM simulations were evaluated to be more accurate and faster than the equivalent modeling with Monte-Carlo simulations. Recently, SpinDoctor [18], a Matlab-based diffusion MRI simulation toolbox that discretizes the Bloch-Torrey equation using finite elements, was released and shown to be hundreds of times faster than Monte-Carlo based

* Corresponding author.
E-mail addresses: vdnguyen@kth.se (V.-D. Nguyen), jjan@kth.se (J. Jansson), jhoffman@kth.se (J. Hoffman), demian.wassermann@inria.fr (D. Wassermann), jingrebecca.li@inria.fr (J.-R. Li).

https://doi.org/10.1016/j.jmr.2019.106611
1090-7807/

simulations at the same level of accuracy for other diffusion sequences.

The discretization of the Bloch-Torrey equation by finite elements is a more recently developed approach for the purpose of dMRI simulations, in contrast to random walk simulations, which have a longer history. While finite element discretization is more difficult to implement than random walk simulations, the approach benefits from long-established theoretical and numerical developments by the mathematical and engineering communities. In particular, software packages for the automated solutions of partial differential equations using finite element discretization, such as FEniCS [19,20], are subject to active support and development. However, because diffusion MRI simulation is a relatively new application area, there is still a gap between the simulation needs of the MRI community and the available tools built on top of these finite element software packages.

The deployment of FEniCS containers [21] opens a new direction to improve productivity and sharing in the scientific computing community. In particular, it can dramatically improve the accessibility and usability of high-performance computing (HPC) systems. In this paper, we address two potential difficulties in using FEniCS for diffusion MRI simulation. First, we simplified software installation by the use of FEniCS containers that are completely portable across multiple platforms. Second, we provide a simulation framework written in Python and whose code is open source. This simulation framework can be seamlessly integrated with cloud computing resources such as Google Colaboratory notebooks (working on a web browser) or with Google Cloud Platform with MPI parallelization.

One of the advantages of the simulation framework we propose here over the Matlab-based SpinDoctor [18] is that the Python code is free, whereas SpinDoctor requires the purchase of the software Matlab. Many researchers are now adopting Python since it is a free, cross-platform, general-purpose and high-level programming language. Plenty of Python scientific packages are available with extensive documentation such as SciPy for fundamentals of scientific computing, NumPy for large and multi-dimensional arrays and matrices, SymPy for symbolic computation, IPython for the enhanced interactive console, Pandas for data structures & analysis, Matplotlib for comprehensive 2D plotting. In addition, parallel computing for finite elements is relatively easy to implement within FEniCS, thus, this framework has advantages over SpinDoctor for very large scale problems.

The disadvantage of this simulation framework compared to SpinDoctor is the current lack of high-order adaptive time-stepping methods in Python tailored to the kind of ODEs systems coming from finite element discretization, whereas such time-stepping methods are available in Matlab. Thus, in contrast to SpinDoctor where an adaptive, variable order, time-stepping method is used, the time-stepping method in the proposed framework is the θ-method, with a fixed time step size. The θ-method is second-order accurate if θ is chosen to be $\frac{1}{2}$.

The simulation framework we propose meets the following needs:

1. the specification of an intrinsic diffusion tensor and a T_2-relaxation coefficient in each geometrical compartment;
2. the specification of a permeability coefficient on the interface between the geometrical compartments;
3. the periodic extension of the computational domain (assumed a box);
4. the specification of general diffusion-encoding gradient pulse sequences;
5. the simulation of thin-layer and thin-tube media using a discretization on manifolds.

Since this framework is based on FEniCS, packaged as an image, it inherits all functionalities of FEniCS related to mesh generation, mesh adaptivity, finite element matrices construction, linear system solve, solution post-processing and display, as well as the underlying FEniCS computational optimization related to the above tasks. Finally, the framework is conceived with cloud computing and high performance computing in mind, thus, it.

1. supports Cloud Computing with Google Colaboratory and Google Cloud Platform;
2. allows for MPI parallelization.

The paper is organized as follows. In Section 2 we recall the diffusion MRI simulation model based on the Bloch-Torrey equation. Then, we propose a portable simulation framework in Section 3 for which the numerical validation and the comparison are carried out in Section 4. Several simulations examples are shown in Section 5. We share some perspectives about the proposed framework in Section 6. The paper is finalised with a conclusion in Section 7.

2. Theory

The evolution of the complex transverse magnetization $U(\boldsymbol{x}, t)$ over time t can be described by the Bloch-Torrey equation [22]. For simplicity we consider a medium composed of two compartments, $\Omega = \Omega_0 \cup \Omega_1$, each of which may be disconnected (see Fig. 1a). The equation takes the following form

$$\frac{\partial U(\boldsymbol{x}, t)}{\partial t} = -i\gamma f(t)\boldsymbol{g}\cdot\boldsymbol{x}U(\boldsymbol{x}, t) - \frac{U(\boldsymbol{x}, t)}{T_2(\boldsymbol{x})} + \nabla\cdot(\boldsymbol{D}(\boldsymbol{x})\nabla U(\boldsymbol{x}, t)), \qquad (1)$$

where i is the complex unit ($i^2 = -1$), $\gamma = 2.67513 \times 10^8$ rad s^{-1} T^{-1} is the gyromagnetic ratio of the water proton, and $\boldsymbol{g}$ is the diffusion gradient including gradient strength $g = \|\boldsymbol{g}\|$ and gradient direction

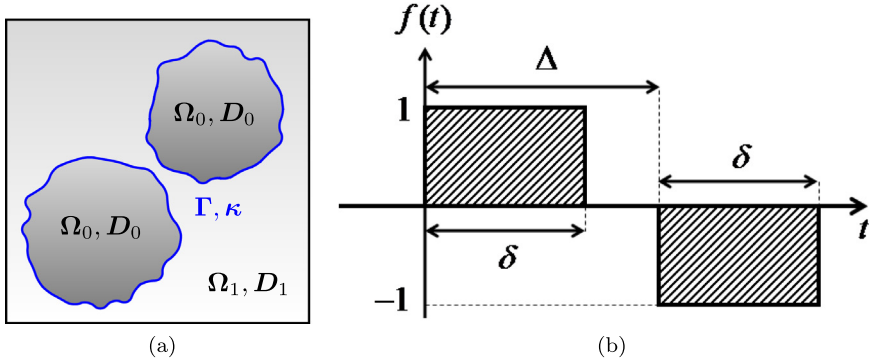


Fig. 1. A composed domain $\Omega = \Omega_0 \cup \Omega_1$ (a), and a PGSE sequence (b).

$\boldsymbol{q} = \frac{\boldsymbol{g}}{\|\boldsymbol{g}\|}$. In the general case, $\boldsymbol{D}(\boldsymbol{x})$ is the diffusion tensor, a symmetric positive definite 3×3 matrix. T_2 relaxation is the process by which the transverse magnetization decays or dephases.

On the interface Γ between different compartments the magnetization is allowed to be discontinuous via the use of a permeability coefficient κ [23]

$$\begin{aligned} &[\![\boldsymbol{D}\nabla U \cdot \boldsymbol{n}^0]\!] = 0, \\ &\{\boldsymbol{D}\nabla U \cdot \boldsymbol{n}^0\} = -\kappa [\![U]\!], \end{aligned} \tag{2}$$

for $\boldsymbol{x} \in \Gamma = \partial\Omega_0 \cap \partial\Omega_1$ and $\boldsymbol{n}^k$ is a normal vector to the interface Γ and pointing outward the volume Ω_k. Here $\{\cdot\}$ and $[\![\cdot]\!]$ denote the average and the jump operators defined on the interface Γ, i.e.

$$\{a\} = \frac{a_0 + a_1}{2}, \quad [\![a]\!] = a_0 - a_1.$$

The temporal profile $f(t)$ can vary for different applications and the most commonly used diffusion-encoding sequence in diffusion MRI literature is called the Pulsed-Gradient Spin Echo (PGSE) sequence [24]. For this sequence, one can write $f(t)$ in the following way (see also Fig. 1b):

$$f(t) = \begin{cases} 1, & 0 \leqslant t \leqslant \delta, \\ -1, & \Delta < t \leqslant \Delta + \delta, \\ 0, & \text{otherwise}. \end{cases} \tag{3}$$

The quantity δ is the duration of the diffusion-encoding gradient pulse and Δ is the time delay between the start of the two pulses. Beyond the PGSE, the Oscillating Gradient Spin Echo (OGSE) [25], nonstandard diffusion sequences such as double diffusion encoding [26–29] and multidimensional diffusion encoding [30] can be modelled.

Concerning the boundary conditions (BCs) on the exterior boundaries $\partial\Omega$, there are two options that are very often employed. One is placing the spins to be simulated sufficiently away from $\partial\Omega$ and impose simple BCs on $\partial\Omega$ such as homogeneous Neumann conditions. This supposes that the spins would have a low probability of having arrived at $\partial\Omega$ during the diffusion experiment. Another option is to place the spins anywhere desired, but to assume that Ω is repeated periodically in all space directions to fill $\mathbb{R}^d$, for example, $\Omega = \prod_{k=1}^{d} [a_k, b_k]$. So, one can mimic the phenomenon where the water molecules can enter and exit the computational domain. Under this assumption of periodic continuation of the geometry, the magnetization satisfies pseudo-periodic BCs on $\partial\Omega$ [8]

$$\begin{aligned} &U_m = U_s e^{i\theta_k(t)}, \\ &\boldsymbol{D}_m \nabla U_m \cdot \boldsymbol{n} = \boldsymbol{D}_s \nabla U_s \cdot \boldsymbol{n}\, e^{i\theta_k(t)}, \end{aligned} \tag{4}$$

where

$$U_m = U(\boldsymbol{x}, t)|_{x_k = a_k}, \quad U_s = U(\boldsymbol{x}, t)|_{x_k = b_k},$$

$$\nabla U_m \cdot \boldsymbol{n} = \nabla U(\boldsymbol{x}, t) \cdot \boldsymbol{n}\,|_{x_k = a_k}, \quad \nabla U_s \cdot \boldsymbol{n} = \nabla U(\boldsymbol{x}, t) \cdot \boldsymbol{n}\,|_{x_k = b_k},$$

and

$$\theta_k(t) := \gamma\, g_k\, (b_k - a_k)\, \mathcal{F}(t), k = 1, \cdots, d, \quad \mathcal{F}(t) = \int_0^t f(s)\, ds.$$

Here we use 'm' and 's' to indicate master and slave components of the pseudo-periodic BCs. The master-slave method corresponds to the implementation of the periodic BCs [31].

The MRI signal S is the total transverse magnetization $U(\boldsymbol{x}, t)$ over Ω measured at the echo time T

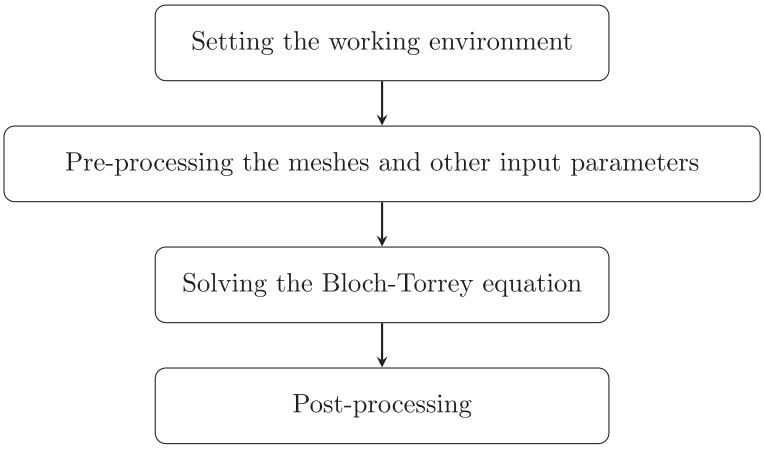
$$S = \int_{\boldsymbol{x} \in \Omega} U(\boldsymbol{x}, T)\, d\boldsymbol{x}. \tag{5}$$

The signal is usually plotted against the gradient strength $g = \|\mathbf{g}\|$ or a quantity called the b-value which is defined as

$$b = \gamma^2 \|\mathbf{g}\|^2 \int_0^T \mathcal{F}(s)^2\, ds. \tag{6}$$

3. Method

For software portability, we consider two container technologies which are Docker [32] and Singularity [33]. They allow for bundling the whole collection of software packages that a user needs in a single file, that can be shared and used by collaborators. This would make a huge impact in scientific applications, where reproducibility is a core concern [34]. In particular, this enables us to develop software that other users can easily test. A software update reduces to a matter of downloading the newest version of a single file and different versions can coexist next to each other for easy consistency checks. We choose Docker for the IPython notebooks and Singularity for the deployment on HPC infrastructure. They follow the same workflow as the following.



3.1. Diffusion MRI simulation library

The solution of the Bloch-Torrey equation and other functionalities related to diffusion MRI simulations have been packaged into the Python library `DmriFemLib`, saved in GitHub.https://github.com/van-dang/DMRI-FEM-Cloud/blob/master/DmriFemLib.py.

Due to considerations related to the way FEniCS envisions the PDE solution workflow, and the fact that the PDE from the diffusion MRI simulation problem has some important differences from the typical PDEs for which FEniCS was designed, we made the following choices regarding the implementation of the numerical method that are different than the choices made in the Matlab-based toolbox SpinDoctor. These choices are:

1. the permeable interface conditions are imposed by the use of the partition of unity finite element method (PUFEM) [35,16];
2. the pseudo-periodic BCs coming from the periodic extension of the computational box are imposed on either side of the box face by a PDE transformation;
3. in case of a non-periodic mesh, the necessary pseudo-periodic BCs are imposed by using an artificially permeability coefficient on the box face whose magnitude is inversely proportional to the finite element mesh size [15,16];
4. the implicit Crank-Nicolson method is chosen as the time-stepping method [13]. It is especially important to ensure the stability with the use of the artificial permeability coefficient;

3.2. Mesh generation

Dealing with meshes is one of the most challenging problems in FEM and we inherit what has been done in Python and FEniCS regarding this issue. For simple geometries, one can internally use some built-in meshes. Meshing a box $\Omega = [0, 10]^3$ with given resolutions `nx`, `ny`, `nz` is simply done by the following commands.

```
nx, ny, nz = 10, 10, 10
mesh = BoxMesh(Point(0.0, 0.0, 0.0), Point(10.0, 10.0, 10.0), nx, ny, nz)
```

For more complicated geometries, it is recommended to use `mshr` [36], the mesh generation component of FEniCS, to generate simplicial DOLFIN meshes in 2D and 3D from geometries described by Constructive Solid Geometry or from surface files, utilizing CGAL and Tetgen as mesh generation backends. The commands below are used to generate a two-layered disk:

```
from mshr import *
R1, R2 = 5, 10; origin = Point(0.,0.);
circle = Circle(origin, R1, segments=32)
domain = Circle(origin, R2, segments=32)
domain.set_subdomain(1, circle)
mesh = generate_mesh(domain, 15) # 15 is the resolution
```

More generally, our framework accepts meshes in the DOLFIN XML format [37]. In this paper, the meshes were generated with GMSH [38], Salomé [39], and ANSA [40]. The GMSH script (`.geo`) and Salomé script (`.py`) are available at https://github.com/van-dang/DMRI-FEM-Cloud/tree/mesh and they are distributed through examples discussed later in the paper. GMSH can be embedded in our framework.

```
import os
# define mesh_name ...
os.system('gmsh -3 '+mesh_name+'.geo -o '+mesh_name+'.msh')
```

All the formats need to be converted to DOLFIN XML format by the use of either `dolfin-convert` available with FEniCS or MESHIO [41]. To convert a mesh from `.msh` to `.xml` in a Colaboratory notebook, we just simply call.

```
import os
os.system("dolfin-convert mesh.msh mesh.xml")
```

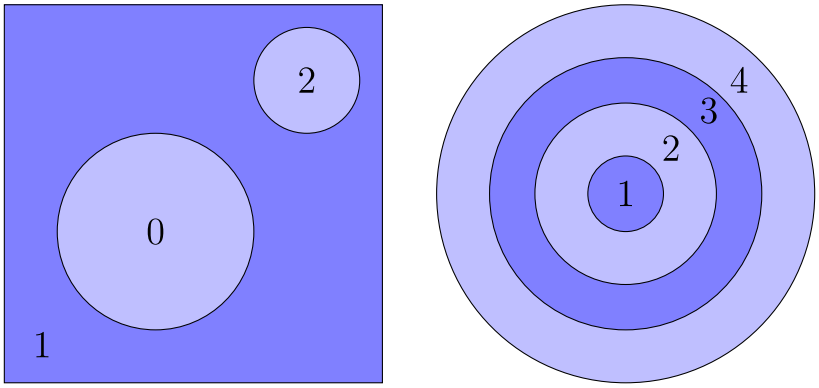


Fig. 2. For multi-compartment domains, the compartments need to be sorted into two groups, `oddgroup` (blue) marked with odd numbers and `evengroup` (light-blue) marked with even numbers which are referred to as the `partition_marker` such that in each group, the compartments should be completely disconnected. It is, therefore, enough to use a `phase` function with two values 0 and 1 to impose the permeability. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

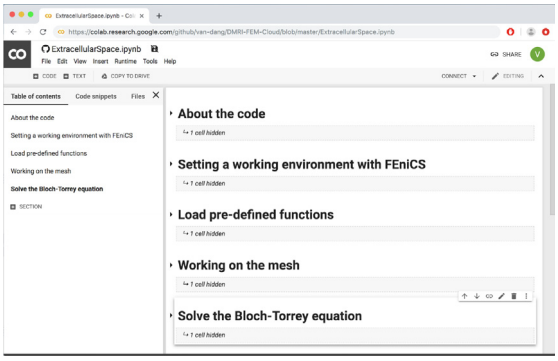


Fig. 3. A typical Google Colaboratory notebook for diffusion MRI simulation.

To impose the interface conditions between compartments, we use a `phase` function which is Φ_h in Eq. (A.6). This function initially supports two-compartment domains since it has only two values 0 and 1. To apply for a multi-compartment domain, the compartments need to be sorted into two groups `oddgroup` and `evengroup`. The permeability is imposed at the intermediate interfaces between the two groups. In each group, however, there is no interface between two compartments or in other words, they are completely disconnected (see Fig. 2).

We defined a routine called `CreatePhaseFunc` to create the `phase` function and the `partition_marker`. If the sub-meshes corresponding to compartments are given, these two functions can be created as follows.

```
# Download the meshes
mesh = Mesh("multi_layer_torus.xml")
cmpt_mesh = Mesh("multi_layer_torus_compt1.xml")
evengroup = []
oddgroup = [cmpt_mesh]
phase, partion_list, partition_marker = CreatePhaseFunc(mesh, evengroup,
    ↪ oddgroup, None)
```

For a multi-compartment domain, a `partition_marker` is used to assign each compartment to an identity that allows for defining nonuniform initial conditions, discontinuous diffusion tensors and discontinuous T_2-relaxation. It is a `MeshFunction` in FEniCS defined as the following.

The `partition_marker` can be generated and saved to a DOLFIN XML file, for instance `partition_marker.xml`. The file structure below shows that the elements (cells) with indices of 0 and 1 are assigned with partition marker 3 and 4 respectively.

```
partition_marker = MeshFunction("size_t", mesh, mesh.topology().dim())
for cell in cells(mesh):
    partition_marker[cell.index()] = <an identity>;
```

```
<?xml version="1.0"?>
<dolfin xmlns:dolfin="http://fenicsproject.org">
  <mesh_function>
    <mesh_value_collection type="uint" dim="3" size="6614">
      <value cell_index="0" local_entity="0" value="3" />
      <value cell_index="1" local_entity="0" value="4" />
      ...
    </mesh_value_collection>
  </mesh_function>
</dolfin>
```

In case the partition markers are defined with GMSH by the use of "physical groups", the file can be simply generated by calling our built-in routine `GetPartitionMarkers`

```
GetPartitionMarkers("mesh.msh", "partition_marker.xml")
```

With a given `partition_marker.xml`, the phase function is generated by.

```
File("partition_marker.xml")>>partition_marker
phase, partition_list = CreatePhaseFunc(mesh, [], [] partition_marker)
```

It requires extra care to generate periodic meshes to use the strong imposition of the pseudo-periodic BCs (see A.1). GMSH supports this by `Periodic` mapping which is equivalent to `Projection` routine in Salomé. As part of the framework, we developed the scripts to generate periodic meshes with cells are available in GMSH https://github.com/van-dang/DMRI-FEM-Cloud/blob/mesh/CirclesInSquare.geo and in Salomé https://github.com/van-dang/DMRI-FEM-Cloud/blob/mesh/SpheresInBox.py.

3.3. The main workspace

The workflow is carried out in the main workspace which is either web-based Jupyter notebooks or a script-based interface. The library `DmriFemLib` and other functionalities need to be loaded here.

```
import os
os.system("wget https://raw.githubusercontent.com/van-dang/
    ↪ DMRI-FEM-Cloud/master/DmriFemLib.py")
from DmriFemLib import *
```

3.3.1. Python notebooks

Google Colaboratory [42] is a free Jupyter notebook environment that requires no setup and runs entirely in the cloud. It can connect to either a hosted runtime provided by Google Cloud or a local runtime. The hosted runtime allows us to access free resources for up to 12 h at a time and the current one, used to obtain the results presented in this paper, has the following configuration:

- Operating system: Ubuntu 18.04.2 LTS
- Processors: 2 x Intel(R) Xeon(R) CPU @ 2.30 GHz
- RAM: 13 GB

In order to check the configuration you can run the following commands.

```
!cat /proc/meminfo #check RAM memory
!lscpu #check processor
!cat /etc/lsb-release #check distribution
```

Fig. 3 shows a typical structure of our Google Colaboratory notebooks where the simulations can run directly since the setup of the FEniCS environment is done within the notebook.

The installation of FEniCS is quite straightforward in the hosted runtime. The command lines are just the same as the installation on Ubuntu.

```
!apt-get install software-properties-common
!add-apt-repository ppa:fenics-packages/fenics
!sudo apt-get update
!apt-get install --no-install-recommends fenics
from dolfin import *; from mshr import *
```

For longer executions, it is more convenient to connect to the local runtime. To this end, one can execute the following command lines to create a local runtime to which the notebook can connect. This command creates a Docker container from the latest stable FEniCS version at the time of writing, given with the `fenics_tag` variable. Inside this container, we install a Jupyter extension developed by Google Colaboratory's developers and then run a Jupyter notebook from within the container on port `8888`.

```
fenics_tag=2019.1.0.r3 # version of FEniCS image
docker run --name notebook-local -w /home/fenics -v
    ↪ $(pwd):/home/fenics/shared -ti -d -p 127.0.0.1:8888:8888
    ↪ quay.io/fenicsproject/stable:${fenics_tag} "sudo pip install
    ↪ jupyter_http_over_ws; sudo apt-get install -y gmsh; jupyter
    ↪ serverextension enable --py jupyter_http_over_ws;
    ↪ jupyter-notebook --ip=0.0.0.0
    ↪ --NotebookApp.allow_origin='https://colab.research.google.com'
    ↪ --NotebookApp.port_retries=0 --NotebookApp.allow_root=True
    ↪ --NotebookApp.disable_check_xsrf=True --NotebookApp.token=''
    ↪ --NotebookApp.password='' --port=8888"
```

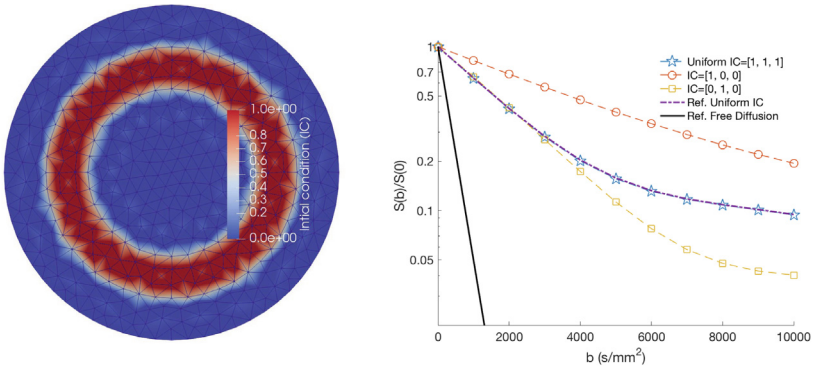
3.3.2. Script-based interface

For the script-based interface with parallel executions, the workspace is available at https://github.com/van-dang/DMRI-FEM-Cloud/blob/master/GCloudDmriSolver.py.

Users can pre-process the inputs for one- and multi-compartment domain by respectively using the functions implemented at https://github.com/van-dang/DMRI-FEM-Cloud/blob/master/PreprocessingOneCompt.py https://github.com/van-dang/DMRI-FEM-Cloud/blob/master/PreprocessingMultiCompt.py.

This workspace can work with Docker image by launching the following command from a Mac or Linux terminal.

```
docker run -ti -p 127.0.0.1:8000:8000 -v $(pwd):/home/fenics/shared -w
    ↪ /home/fenics/shared quay.io/fenicsproject/stable:current
```



(a) Initial conditions [0, 1, 0] (b) Signals

Fig. 4. Simulations of diffusion inside a three-layered disk of $[5, 7.5, 10]$ μm with different settings of the initial conditions for a PGSE sequence with $\Delta = 43100$ μs and $\delta = 10600$ μs. (a) show the initial conditions with [0, 1, 0]. The signals are strongly dependent on how the initial conditions are set up (b). The accuracy of the simulations is verified by comparing with the reference signal for the uniform distribution of the initial conditions.

However, in the HPC context, Singularity is preferable to Docker due to the security, the accessibility, the portability, and the scheduling issues [33]. Fortunately, it is straightforward to build a Singularity image from a Docker image and for our framework, the command lines are as follows.

```
wget https://raw.githubusercontent.com/van-dang/DMRI-FEM-Cloud/
    ↪ singularity_images/Singularity_recipe_FEniCS_DMRI
sudo singularity build -w writable_fenics_dmri.simg
    ↪ Singularity_recipe_FEniCS_DMRI
```

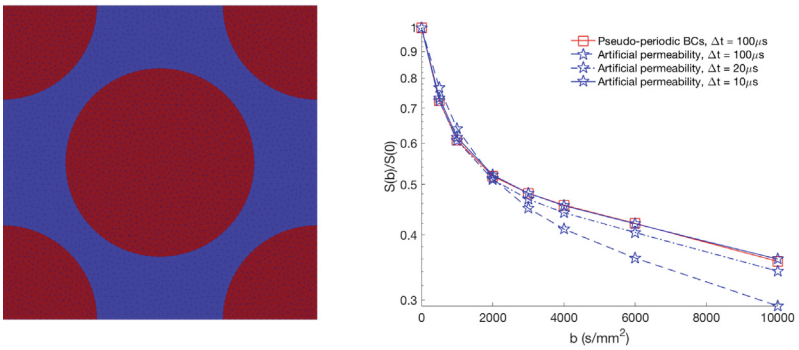
3.3.3. Code structure

Although the interfaces are different between the web-based and the script-based workspaces, they have similar structures with three main classes.

- `MRI_parameters` manages the diffusion pulses such as sequence type, *b*-values, *g*-value and the conversion between them.
- `MRI_domain` manages the finite element meshes, function spaces, domain sizes, diffusion tensors, permeability, and boundary markers.
- `MRI_simulation` manages the initial conditions, time-stepping sizes, linear solvers, solutions, and post-processing.

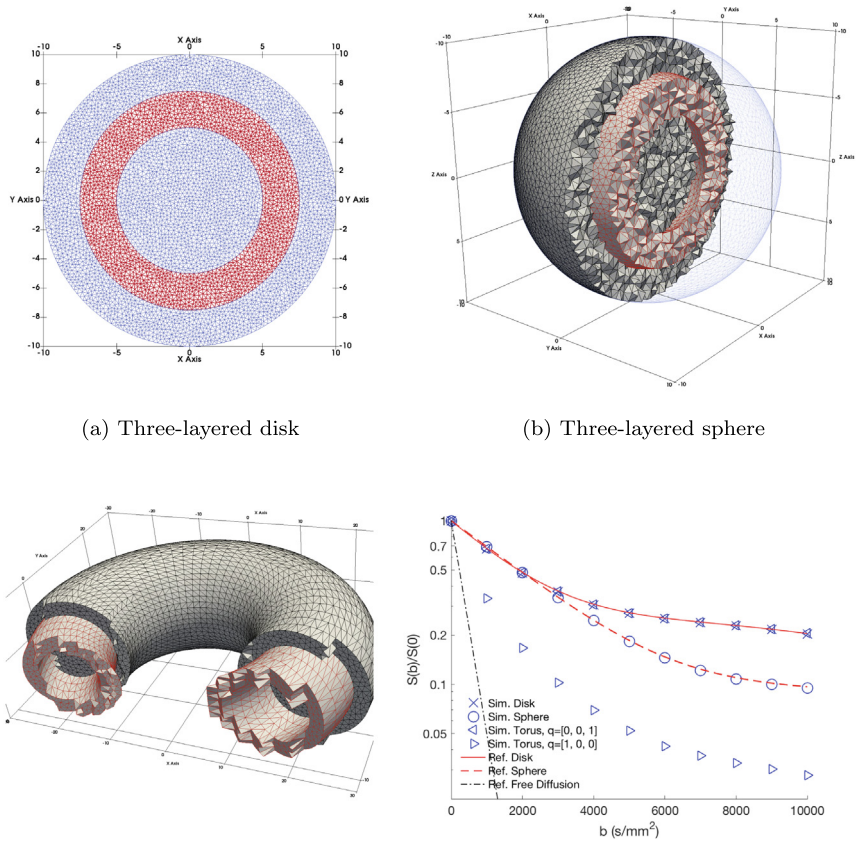
In `MRI_domain`, the boolean variable `IsDomainMultiple` is used to switch between the single-compartment and the multi-compartment domains. Both strong and weak imposition of the pseudo-periodic BCs have some advantages and disadvantages. The strong imposition works efficiently with periodic meshes with higher accuracy compared to the weak imposition. In some cases, it is, however, not practical to generate periodic meshes. We allow for both options by the use of a boolean variable `IsDomainPeriodic`. When it is `True`, Eq. (A.1) is solved, otherwise, Eq. (1) is solved.

In what follows, we show how to define an arbitrary diffusion sequence and how to use `partition_marker` to define some input parameters on heterogeneous domains.



(a) A square box with circular cells (b) Normalized signals

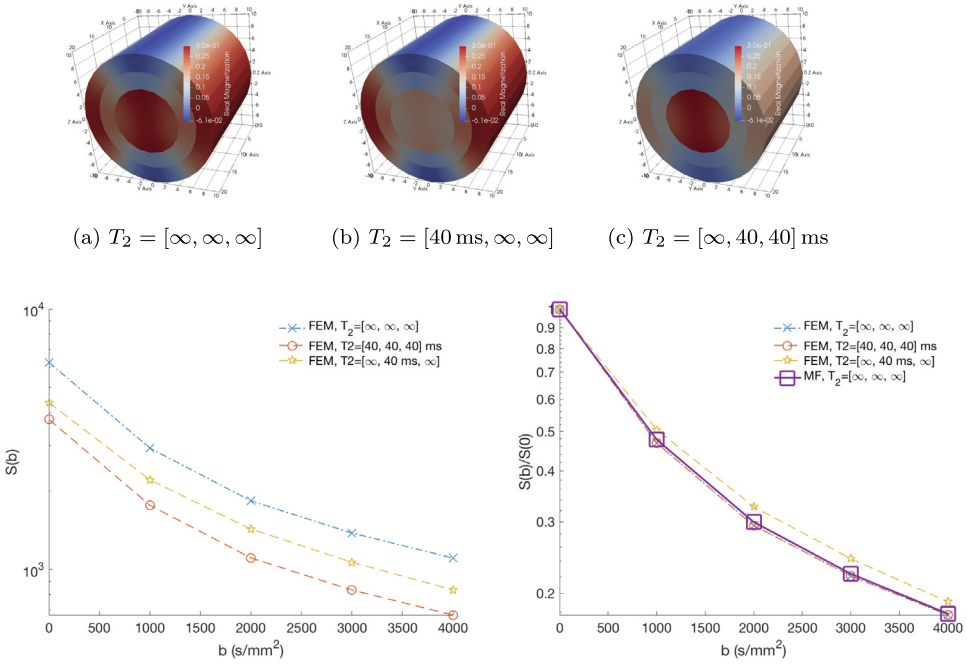
Fig. 5. The artificial permeable method approaches the pseudo-periodic BCs. The time-step size needs to be small to achieve the same accuracy. It is, however, important to recall that the artificial permeable method is useful for non-periodic computational boxes.



(a) Three-layered disk (b) Three-layered sphere

(c) Three-layered torus (d) Signals

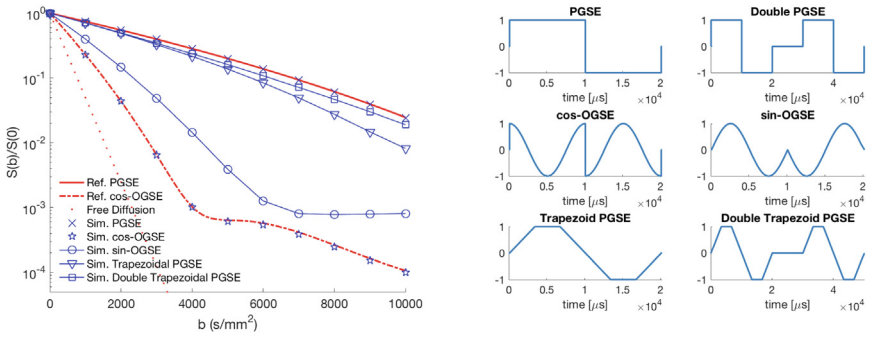
Fig. 6. Three-layered structures and their signals for a PGSE with $\Delta = 43100\ \mu s, \delta = 10600\ \mu s$. The time step size is $\Delta t = 200\ \mu s$.



(a) $T_2 = [\infty, \infty, \infty]$ (b) $T_2 = [40\,\text{ms}, \infty, \infty]$ (c) $T_2 = [\infty, 40, 40]\,\text{ms}$

(d) Signals

Fig. 7. T_2-effects of diffusion inside a three-layered cylinder for a PGSE with $\Delta = \delta = 10$ ms, permeability $\kappa = 10^{-5}$ m/s to the magnetization at $b = 4000$ s/mm^2 (a, b, c), and to the signals for b between 0 and 4000 s/mm^2 (d).



(a) Signals (b) Temporal profiles

Fig. 8. Simulated signals inside a disk of radius 5 μm (a) for different temporal profiles: PGSE, Double PGSE, cos-OGSE, sin-OGSE, Trapezoidal PGSE, and Double Trapezoidal PGSE with $\delta = \Delta = 10000$ μs (b). The simulated signals match very well the reference signals for the PGSE and cos-OGSE. The signals with OGSE sequences decay faster compared to the others.

3.3.4. General diffusion-encoding sequence

The framework allows for arbitrary temporal profiles $f(t)$. During the simulation, we need to compute its integral $\mathcal{F}(t) = \int_0^t f(s)\,ds$ and convert between b-value and the gradient strength g-value following Eq. (6). `SymPy` is used to compute the symbolic integration. So, users only need to provide the expression of $f(s)$ to the function member `fs_sym` of the class `MRI_parameters()`. $\mathcal{F}(t)$ and the conversion between b-value, g-value are automatically done. For example, a cos-OGSE sequence

$$f(s) = \begin{cases} \cos(\omega s), & \text{if } s \leqslant \delta, \\ -\cos(\omega(s-\tau)), & \text{if } \tau < s \leqslant \delta + \tau, \\ 0, & \text{otherwise}, \end{cases} \tag{7}$$

with $\omega = \frac{2n\pi}{\delta}, \tau = \frac{\delta+\Delta}{2}$ can be simply defined as the following

```
import sympy as sp
mp = MRI_parameters()
...
mp.delta, mp.Delta = 10000, 10000
mp.T = mp.delta+mp.Delta
mri_para.nperiod = 2
omega = 2.0*mri_para.nperiod*pi/mri_para.delta
tau = mp.T/2.
mri_para.fs_sym = sp.Piecewise(
    (  sp.cos(omega*mri_para.s) ,    mri_para.s <= mri_para.delta ),
    (  0.,                           mri_para.s <= tau ),
    (  -sp.cos(omega*(mri_para.s-tau)), mri_para.s <= mri_para.delta +
        ↪ tau ),
    (  0., True )
)
...
mp.Apply() # F(t) and the conversion between b and q are done here
```

Table 1
Timing in minutes of neuron simulations on Google Colaboratory for a PGSE sequence with $\Delta = 43100$ μs, $\delta = 10600$ μs and different time step sizes Δt.

Neuron	Mesh size	$\Delta t = 50$ μs	$\Delta t = 100$ μs
04b_pyramidal7aACC	615146 vertices	119	64
25o_spindle17aFI	51792 vertices	21	10
03b_pyramidal2aACC	27811 vertices	6	3

3.3.5. Initial conditions

By default, we initialize the spins to be one everywhere. However, it is possible to define discontinuous initial conditions which are illustrated in the following code snippet.

```
mri_simu = MRI_simulation()
mri_para = MRI_parameters()
mymesh = Mesh(...)
mri_domain = MRI_domain(mymesh, mri_para)
...
IC_array = [0, 1, 0];
dofmap_DG = mri_domain.V_DG.dofmap()
disc_ic = Function(mri_domain.V_DG);
for cell in cells(mymesh):
    cmk = partition_marker[cell.index()]
    cell_dof = dofmap_DG.cell_dofs(cell.index())
    disc_ic.vector()[cell_dof] = IC_array[cmk];
disc_ic=project(disc_ic, mri_domain.V)
...
mri_simu.solve(mri_domain, mri_para, linsolver, disc_ic)
```

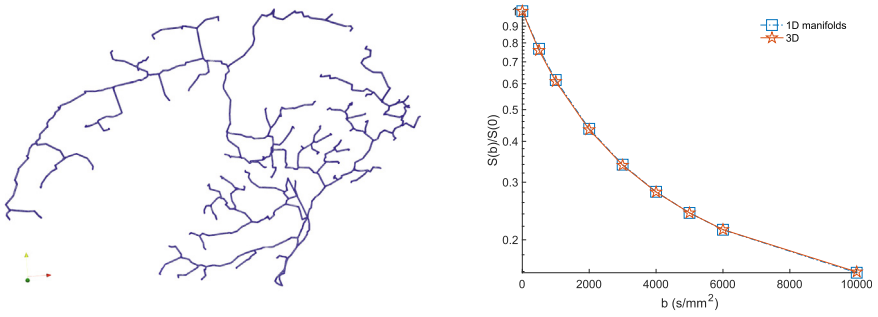


Fig. 9. A comparison between signals inside a neuron from the drosophila melanogaster for a standard 3D mesh and the corresponding 1D manifolds. For $\Delta t = 200$ µs, it costs only 3 s for 1D manifolds but 380 s for 3D to compute the signal for one b-values with the same accuracy.

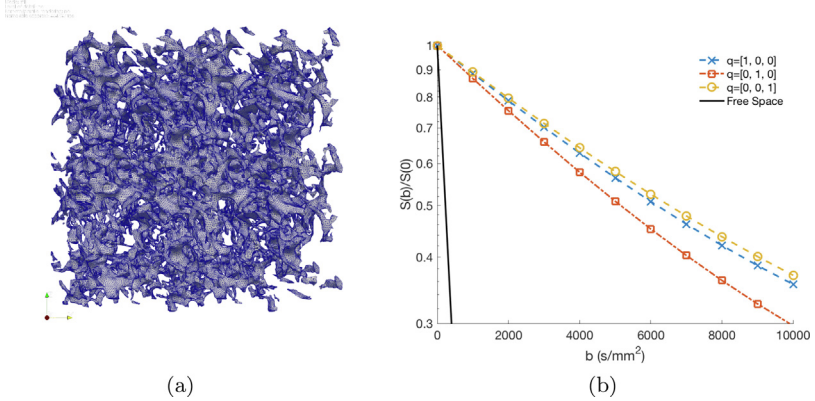


Fig. 10. The mesh of an extracellular space with 462420 vertices and 926058 tetrahedrons (a). It was reconstructed from the medical segmentation published at http://synapseweb.clm.utexas.edu/2013kinney (see also [46]). Two directions in the xz-plane give quite similar signals showing that the domain is quite isotropic in these directions and they both are distinguishable to the signals in the y-direction (b).

3.3.6. Diffusion coefficients and tensors

We allow for a general definition of the diffusion tensor with $d \times d$ components

$$\boldsymbol{D}(\boldsymbol{x}) = \left[d_{jk}(\boldsymbol{x})\right]_{j=1..d,k=1..d}, \tag{8}$$

where $d_{jk}(\boldsymbol{x})$ is cell-based piecewise continuous. We loop through all elements (cells) and the value can be determined by the coordinates of the cell midpoint `p = cell.midpoint()` or a given `partition_marker`.

$D = 3 \times 10^{-3}$ mm^2/s. The membrane between the compartments, if any, is permeable with the permeability of $\kappa = 10^{-5}$ m/s. The simulated signals are compared to the reference ones computed by the matrix formalism (MF) method [45].

We provide a complete simulation method of diffusion inside the multilayered structures such as concentric disks, cylinders, spheres, and torus with the mesh generation software GMSH [38].

First, we study diffusion inside a three-layered disk of $[5, 7.5, 10]$ μm with different settings of the initial conditions.

```
partition_marker = MeshFunction("size_t", mesh, mesh.topology().dim())
# define partition markers
...
D0_array=[3e-3, 1e-3, 3e-3]
# Variable diffusion tensor
V_DG=mri_domain.V_DG; dofmap_DG = V_DG.dofmap();
d00 = Function(V_DG); d01 = Function(V_DG); d02 = Function(V_DG)
d10 = Function(V_DG); d11 = Function(V_DG); d12 = Function(V_DG)
d20 = Function(V_DG); d21 = Function(V_DG); d22 = Function(V_DG)
for cell in cells(mymesh):
    p = cell.midpoint() # the coordinate of the cell center.
    cmk = partition_marker[cell.index()]
    cell_dof = dofmap_DG.cell_dofs(cell.index())
    d00.vector()[cell_dof] = D0_array[cmk];
    d11.vector()[cell_dof] = D0_array[cmk];
    d22.vector()[cell_dof] = D0_array[cmk];
mri_domain.ImposeDiffusionTensor(d00,d01,d02,d10,d11,d12,d20,d21,d22)
```

3.3.7. T_2-relaxation coefficient

By default T_2-relaxation is set to be `1e16`. However, users can define it similarly to the diffusion entry. The following code shows how to define T_2 for a three-compartment domain.

Fig. 4a shows the setting `IC_array=[0, 1, 0]` as discussed in Section 3.3.5. The solver is available at https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/DiscontinuousInitialCondition.ipynb.

```
T2_array=[4e16, 4e4, 4e4]
dofmap_DG = mri_domain.V_DG.dofmap()
T2 = Function(mri_domain.V_DG);
for cell in cells(mymesh):
    cmk = partition_marker[cell.index()]
    cell_dof = dofmap_DG.cell_dofs(cell.index())
    T2.vector()[cell_dof] = T2_array[cmk];
mri_para.T2 = T2
```

3.4. Solution visualization and post-processing

After solving the Bloch-Torrey equation, the solutions are saved, visualized and the signals are computed following Eq. (5) in a routine called `PostProcessing`. Matplotlib [43] is used for simple visualizations. For more advanced features, Paraview [44] can be externally used for the saved solutions.

4. Numerical validation and comparison

Unless stated otherwise, the simulations were performed for a PGSE with $\Delta = 43100$ μs, $\delta = 10600$ μs, b-values between 0 and 10000 s/mm^2, and the diffusion coefficient of

The time step size of $\Delta t = 200$ μs is used. The signals are shown in Fig. 4b where we show that the signals are strongly dependent on how we set up the initial conditions. The accuracy of the simulations is verified by comparing with the reference signal for the uniform distribution of the initial conditions.

To reduce the computational domain, one can assume that the domain is periodically repeated and our framework provides two options for this purpose. With a periodic mesh, it is strongly recommended to solve Eq. (A.1) with periodic boundary conditions (Eq. A.2). However, it is usually challenging to generate such a periodic mesh. So, we also provide an approximation using an artificial permeability coefficient [15,16] (see also in B) for non-periodic meshes. To illustrate this capacity, we consider a square

$\Omega = [-5\ \mu m, 5\ \mu m]^2$ including some permeable periodic cells with the permeability $\kappa = 10^{-5}$ m/s (see Fig. 5a). The signals were computed for a PGSE with $\Delta = 13000\ \mu s, \delta = 10000\ \mu s, \boldsymbol{q} = \frac{[1,1,0]}{\sqrt{2}}$ (see Fig. 5b). We see that the artificial-permeability method approaches the pseudo-periodic BCs. To achieve the same accuracy, the artificial-permeability method needs $\Delta t = 10\ \mu s$ which is ten times as smaller as the periodic BCs. The solver is available at https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/PeriodicDomains.ipynb.

We now consider discontinuous diffusion tensors mentioned in Section 3.3.6 to study diffusion in three-layered structures including a disk, a sphere and a torus whose radii are $5, 7.5$ and $10\ \mu m$ respectively (see Fig. 6a, b, and c). For the torus, the radius from the center of the hole to the center of the torus tube is $R = 20\ \mu m$. The simulated signals match very well to the reference signals with the time step size of $\Delta t = 200\ \mu m$ (see Fig. 6d). The Python source code is available at https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/MultilayeredStructures.ipynb.

The three-layered cylinder is again used to illustrate the effect of T_2-relaxation to the magnetization and the signal attenuation. The gradient direction is $\boldsymbol{q} = [0, 1, 0]$ which is perpendicular to the cylinder axis. As expected, the transverse magnetization decays faster for smaller T_2 (Fig. 7a, b, and c). The signals $S(b)$ are quite different when T_2 varies (Fig. 7d). Here we also show that our signals approximate accurately the reference ones calculated by the matrix formalism (solid curve in the figure) [45]. In short, T_2-relaxation can be used as one of the sources of the image contrast. The Python source code is available at https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/T2_Relaxation.ipynb.

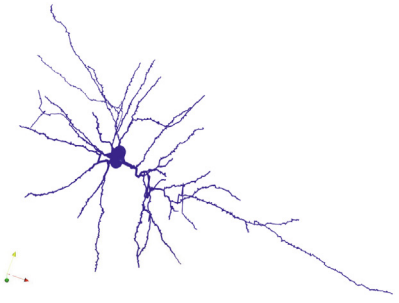
Now, we use the solver to compare the signals inside a disk of radius $5\mu m$ for some temporal profiles: PGSE, Double PGSE, cos-OGSE, sin-OGSE, Trapezoidal PGSE, and Double Trapezoidal PGSE with $\delta = \Delta = 10000\ \mu s$ (see Fig. 8b). The solver is available at https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/ArbitraryTimeSequence.ipynb.

The simulated signals match very well the reference signals for the PGSE and cos-OGSE. The signals with OGSE sequences decay faster compared to the others (see Fig. 8a).

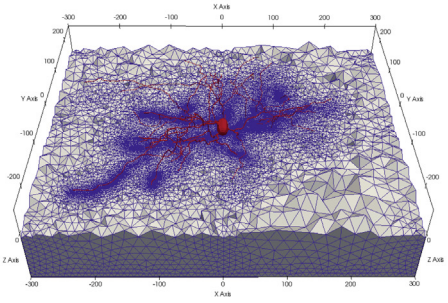
5. Simulation examples

5.1. Realistic neurons

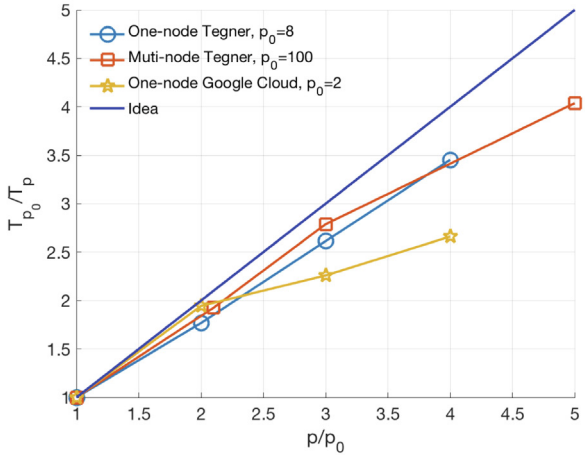
We consider a population of 36 pyramidal and 29 spindle neurons. They are distributed in the anterior frontal insula (aFI) and the anterior cingulate cortex (ACC) of the neocortex of the human brain. They share some morphological similarities such as having a single soma and dendrites branching on opposite sides. This population consists of 20 neurons for each type in aFI, and 9 spindles, 16 pyramidals in ACC. We have published these volume meshes at https://github.com/van-dang/RealNeuronMeshes.



(a) Neuron `04b_pyramidal7aACC`



(b) The mouse neuron embedded in a box



(c) Strong parallel scaling of one-node and multi-node simulations

Fig. 11. The simulation performance is verified for a single computational node on the neuron `04b_pyramidal7aACC` and for multiples nodes on the mouse neuron embedded in a box presented in Section 7.6 in [16]. The strong scaling on Tegner is good both on one node (32 CPUs) and multi-node (scaling up to 500 CPUs). For a small number of cores (2, 4, 6, 8) on Google Cloud, the scaling is less good. It is because the workload and data partition per process are greatly exceeding the ideal work and data per process ratio with the mesh size of 2.5 M tetrahedrons.

The solver is available at https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/RealNeurons.ipynb.

Table 1 shows the computational time in minutes of neuron simulations on Google Colaboratory for a PGSE sequence with $\Delta = 43100\ \mu s, \delta = 10600\ \mu s, b = 4000\, s/mm^2$ and two different time step sizes $\Delta t = 50, 100\ \mu s$. The relative difference in signals between them is within 4%. With $\Delta = 100\ \mu s$ it costs about an hour for the largest neuron with 615146 vertices whereas it costs only 3 min with a small neuron with 27811 vertices.

In addition to the standard approach using volume elements, we also allow for simulating on manifolds following the method developed in [17]. Fig. 9 shows a comparison between signals inside a neuron from the drosophila melanogaster for a standard 3D mesh and the corresponding 1D manifolds. For $\Delta t = 200\mu s$, it costs only 3 s for 1D manifolds but 380 s for 3D to compute the signal for one b-values with the same accuracy. For more details, it is recommended to look at the solver available at https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/Manifolds.ipynb.

5.2. Extracellular space

It is challenging to perform simulations on extracellular space (ECS) due to the geometrical complexity. The thickness of ECS is tiny compared to the computational domain. It is extremely time-consuming to use Monte-Carlo approaches. If the reflection condition is applied, the particle undergoes multiple reflections until no further surface intersections are detected, and if the rejection method is applied, the time step sizes need to be very small to be accurate.

In this section, we show that it is efficient to use our framework. We tested with the ECS extracted from the medical segmentation published at http://synapseweb.clm.utexas.edu/2013kinney (see also [46]). The volume mesh is shown in Fig. 10a with 462420 vertices and 926058 tetrahedrons The processed meshes are available in the following link https://github.com/van-dang/DMRI-FEM-Cloud/raw/mesh/2E_ExtraCellular_group_10um_vol.xml.zip.

The Google Colab-based solver is available in the following link https://colab.research.google.com/github/van-dang/DMRI-FEM-Cloud/blob/master/ExtracellularSpace.ipynb.

The timing per b-value is about 30 min on Google Colaboratory for the time discretization $\Delta t = 1$ ms. With half of this time-step size, i.e., $\Delta t = 500\ \mu s$, it takes about an hour for $b = 10000\, s/mm^2$ on Google Colaboratory and the difference in the signals compared to $\Delta t = 1000\ \mu s$ is only 1%. The signals for three principle gradient directions are shown in Fig. 10b. Two directions in the xz-plane give quite similar signals showing that the domain is quite isotropic in these directions and they both are distinguishable from the signals in the y-direction.

5.3. Parallelization

Now we verify the simulation performance with the Singularity image on a 12-month free trial of Google Cloud Platform (https://cloud.google.com) and Tegner (PDC - KTH). The script-based interface is used. It shares the core functionalities with the Python Notebook interface and supports all the functionalities discussed in the paper except the artificial permeability implementation which is still in development. So, the mesh needs to be periodic to have the pseudo-periodic BCs in the parallel execution.

First, we show the simulation performance on one computational node on the neuron `04b_pyramidal7aACC` (Fig. 11a) with the mesh size of about 0.6 M vertices (2.5 M tetrahedrons) and the time discretization of $\Delta t = 200\ \mu s$. A PGSE with $\Delta = 43100\ \mu s, \delta = 10600\ \mu s$ is used. The commands to execute the simulation with the FEniCS image are follows.

```
singularity exec -B $PWD writable_fenics_dmri.simg python3
    ↪ PreprocessingOneCompt.py -o onecompt_files.h5
ListNumProcs="2 4 6  8"    # one-node in Google Cloud
ListNumProcs="5 10 15 20" # one-node in Tegner
for p in ${ListNumProcs}
do
    singularity exec -B $PWD writable_fenics_dmri.simg mpirun -n ${p}
        ↪ python3 GCloudDmriSolver.py -f onecompt_files.h5 -M 0 -b 1000
        ↪ -d 10600 -D 43100 -k 200 -K 3e-3 -gdir 1 0 0
done
```

On Tegner with 20 processors, it costs about 7 min per one b-value whereas on Google Cloud with 8 processors, it costs about 30 min.

Then, we verify with 25 computational nodes on the sample presented in Section 7.6 [16]. The sample consists of a pyramidal neuron of an adult female mouse [47] embedded in the center of a computational domain $\Omega = [-300, 300] \times [-250, 250] \times [-100, 100]\ \mu m^3$ (Fig. 11b). We assume that there is a permeable membrane with $\kappa = 10^{-5}$ m/s between the neuron and the extracellular space. The whole mesh (box + neuron) has about 1.5 M vertices (8.5 M tetrahedrons). The neuron itself consists of 131996 vertices and 431326 tetrahedrons. The whole mesh and sub-mesh (neuron) are available for download at https://github.com/van-dang/DMRI-FEM-Cloud/raw/mesh/volume_box_N_18_7_3_5L_fine.xml.zip https://github.com/van-dang/DMRI-FEM-Cloud/raw/mesh/volume_N_18_7_3_5L_fine.xml.zip.

Below are the commands to execute the simulation with the FEniCS image.

```
singularity exec -B $PWD writable_fenics_dmri.simg python3
    ↪ PreprocessingMultiCompt.py -o multcompt_files.h5
ListNumProcs="100 210 300 500" # 25 nodes in Tegner
for p in ${ListNumProcs}
do
    mpirun -n $p singularity exec -B $PWD writable_fenics_dmri.simg
        ↪ python3 GCloudDmriSolver.py -f multcompt_files.h5 -M 1 -b
        ↪ 1000 -p 1e-5 -d 10600 -D 43100 -k 200 -gdir 0 1 0
```

The simulation with 500 processors costs about 20 min per b-value with $\Delta t = 200$ µs.

The strong parallel scaling is shown in Fig. 11c. Ideally, one expects that the timing is reduced by half when the number of the processors is doubled. Assume that we start with p_0 processors and measure the timing T_{p_0}. Then, we increase the number of the processors to p ($p \geqslant p_0$) and measure the timing T_p. In general, the ideal scaling (the blue curve in Fig. 11c) is

$$\frac{T_{p_0}}{T_p} = \frac{p}{p_0}.$$

We now compare between the realistic scaling of our framework and this ideal linear scaling for $p_0 = 8,100$ on Tegner and $p_0 = 2$ on Google Cloud. The strong scaling on Tegner is good both on one node (32 CPUs) and multi-node (scaling up to 500 CPUs). For a small number of cores (2, 4, 6, 8) on Google Cloud, the scaling is less good. It is because the workload and data partition per process are greatly exceeding the ideal work and data per process ratio with the mesh size of 2.5 M tetrahedrons.

6. Discussion

The proposed framework can be viewed as the Python version of the Matlab-based SpinDoctor. As verified in [18] for SpinDoctor, this present framework is supposed to be faster and more accurate than Monte-Carlo simulation packages such as Camino. More importantly, our approach benefits from a long history of theoretical and numerical developments by the mathematical and engineering communities. It enhances software reliability which is one of the core concerns in medical applications. Additionally, this framework inherits all of the PDE solution functionalities of FEniCS. Thus, extensions and generalizations of the present dMRI simulation problem, including the coupling with flow, the simulation on deforming domains like the heart, or the coupling of simulations in manifolds with simulations in 3D domains would be straightforward.

We also focused on advanced software engineering features such as portability and parallelization. As other cloud-based software developments, this framework brings reproducible science and open-source software to computational diffusion MRI. It speeds up the method development process since the results are easy to confirm and new methods can be easily developed on top of the existing methods. New algorithms written as Google Colaboratory notebooks can quickly circulate in the MRI community and this allows for active collaboration between research groups. With the parallelization on supercomputers, the simulations lasting weeks can be now reduced to hours or minutes. It enables us to develop, in the near future, real-time simulations of diffusion MRI in which the computer simulation runs at the same rate as the actual physical system.

Since SpinDoctor couples the finite element discretization with optimized adaptive ODE solvers, it is more efficient than our framework in terms of time discretization. The analogous ODE solvers written in Python can be found in the SciPy Library [48] but they are not ready to use within our framework: they do not yet efficiently support the mass matrix and the sparse Jacobian matrix. The lumped mass matrix approach can be used to fix the first issue but more investigations are needed to resolve the latter issue.

Generating finite element meshes from medical segmentation is very challenging. Complicated surface meshes currently need to be processed outside the framework to obtain a good quality finite element mesh. Streamlining this process is an interesting direction of future investigation and it may be well worthwhile to develop algorithms to automate this process.

7. Conclusions

We proposed a portable simulation framework for computational diffusion MRI that works efficiently with cloud technology. The framework can be seamlessly integrated with cloud computing resources such as Google Colaboratory notebooks working on a web browser or with Google Cloud Platform with MPI parallelization. Many simulation needs of the field were addressed by the use of advanced finite element methods for both single- and multi-compartment diffusion domains, with or without permeable membrane and periodic boundaries. We showed the accuracy, the computational times, and parallel computing capabilities through a set of examples, while mentioning straightforward future extensions. The framework contributes to reproducible science and open-source software in computational diffusion MRI. We hope that it will help to speed up method developments and stimulate research collaborations.

Acknowledgement

This research has been supported by the Swedish Energy Agency, Sweden with the project ID P40435-1; MSO4SC with the Grant No. 731063; the Basque Excellence Research Center (BERC 20142017) program by the Basque Government; the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa accreditation SEV-2013-0323; the ICERMAR ELKARTEK project of the Basque Government; the projects of the Spanish Ministry of Economy and Competitiveness with reference MTM2013-40824-P and MTM2016-76016-R. The research was

conducted on resources provided by the Swedish National Infrastructure for Computing (SNIC) at the Center of High- Performance Computing (PDC). We also would like to thank ANSA from Beta-CAE Systems S. A., who generously provided an academic license. The first author would like to thank Niyazi Cem Degirmenci for his enthusiastic supports.

Appendix A

The methods imposed in our framework are based on the partition of unity finite element method (PUFEM) to manage the interface conditions [16]. Both weak and strong implementation of the periodicity with some advantages and disadvantages are included. The θ-method is used for the time discretization.

A.1. Strong implementation of the pseudo-periodic BCs

The complex-valued and time-dependent term in the pseudo-periodic boundaries make it too difficult to implement in a standard FEM software package. So, one can transform the pseudo-periodic BCs to the periodic ones. Following [10,12], one can choose to transform the magnetization to a new unknown $u(\boldsymbol{x},t)$:

$$u(\boldsymbol{x},t) = U(\boldsymbol{x},t)\, e^{i\,\gamma\,\mathcal{F}(t)\,\boldsymbol{g}\cdot\boldsymbol{x}}.$$

The Bloch-Torrey PDE (1) is then transformed to [12]

$$\frac{\partial u}{\partial t} = -i\gamma\mathcal{F}(\boldsymbol{g}\cdot\boldsymbol{D}\nabla u + \nabla u\cdot\boldsymbol{D}\boldsymbol{g}) - (\gamma\mathcal{F})^2\boldsymbol{g}\cdot\boldsymbol{D}\boldsymbol{g}u - \frac{u}{T_2} + \nabla\cdot(\boldsymbol{D}\nabla u), \tag{A.1}$$

with periodic BCs

$$\begin{aligned} &u_m = u_s, \\ &\boldsymbol{D}_m\nabla u_m\cdot\boldsymbol{n} = \boldsymbol{D}_s\nabla u_s\cdot\boldsymbol{n} \end{aligned} \tag{A.2}$$

The homogeneous Neumann boundary condition of U leads to

$$\boldsymbol{D}\nabla u\cdot\boldsymbol{n} = i\gamma\mathcal{F}u\boldsymbol{D}\boldsymbol{g}\cdot\boldsymbol{n}.$$

The interface conditions (Eq. (2)) are changed to

$$\begin{aligned} &[\![\boldsymbol{D}\nabla u\cdot\boldsymbol{n}_0]\!] = 2i\gamma\mathcal{F}\{u\boldsymbol{D}\boldsymbol{g}\cdot\boldsymbol{n}_0\}, \\ &\{\boldsymbol{D}\nabla u\cdot\boldsymbol{n}_0\} = -\kappa[\![u]\!] + \tfrac{i\gamma\mathcal{F}}{2}[\![u\boldsymbol{D}\boldsymbol{g}\cdot\boldsymbol{n}_0]\!] \end{aligned} \tag{A.3}$$

Since the magnetization is discontinuous ($m_0 \neq m_1$ on the interface), Eq. (A.3) shows the flux is also discontinuous.

Following the same PUFEM approach proposed in [16], we obtain the following weak form

$$\left(\frac{\partial}{\partial t}u, v\right)_{\Omega_0\cup\Omega_1} = F(u,v,t),$$

where

$$\begin{aligned} F(u,v,t) = &-(i\gamma\mathcal{F}(\boldsymbol{g}\cdot\boldsymbol{D}\nabla u + \nabla u\cdot\boldsymbol{D}\boldsymbol{g}), v)_{\Omega_0\cup\Omega_1} \\ &-\left((\gamma\mathcal{F})^2\boldsymbol{g}\cdot\boldsymbol{D}\boldsymbol{g}u + \frac{u}{T_2}, v\right)_{\Omega_0\cup\Omega_1} \\ &-(\boldsymbol{D}\nabla u, \nabla v)_{\Omega_0\cup\Omega_1} + \\ &< -\kappa[\![u]\!] + \frac{i\gamma\mathcal{F}}{2}[\![u\boldsymbol{D}\boldsymbol{g}\cdot\boldsymbol{n}_0]\!], [\![v]\!] >_\Gamma + \\ &< 2i\gamma\mathcal{F}\{u\boldsymbol{D}\boldsymbol{g}\cdot\boldsymbol{n}_0\}, \{v\} >_\Gamma + \\ &< i\gamma\mathcal{F}u\boldsymbol{D}\boldsymbol{g}\cdot\boldsymbol{n}, v >_{\Gamma_0^N\cup\Gamma_1^N}. \end{aligned} \tag{A.4}$$

We consider a partition of the time domain $0 = t_0 < t_1 < \ldots < t_N = T$ associated with the time intervals $I_n = (t^{n-1}, t^n]$ of length $k^n = t^n - t^{n-1}$ and u^n be an approximation of $u(\boldsymbol{x},t)$ for a given a triangulation $\mathcal{T}^h$ at $t = t^n$.

The PUFEM with the time-stepping θ-method is stated as: Find $u_h^n = (u_{h,0}^n, u_{h,1}^n) \in \boldsymbol{V}_h$ such that

$$\begin{aligned} \left(\frac{u_h^n - u_h^{n-1}}{k^n}, v_h\right)_{\Omega_0\cup\Omega_1} = &\,\theta F(u_h^n, v_h, t^n) + (1 \\ &- \theta) F(u_h^{n-1}, v_h, t^{n-1}), \end{aligned} \tag{A.5}$$

for all $v_h = (v_{0,h}, v_{1,h}) \in \boldsymbol{V}_h$, where $(a,b)_{\Omega_{0,h}\cup\Omega_{1,h}} = ((1-\Phi_h)a_0, b_0)_{\Omega_h} + (\Phi_h a_1, b_1)_{\Omega_h}$, and Φ_h is an element-wise constant function:

$$\Phi_h = \begin{cases} 1, & \text{in } \Omega_{1,h}, \\ 0, & \text{in } \Omega_{0,h}. \end{cases} \tag{A.6}$$

The bilinear and linear forms are defined by

$$\begin{aligned} &\boldsymbol{a}(u_h^n, v_h) = \left(\tfrac{u_h^n}{k^n}, v_h\right)_{\Omega_0\cup\Omega_1} - \theta F(u_h^n, v_h, t^n), \\ &\boldsymbol{L}(v_h) = \left(\tfrac{u_h^{n-1}}{k^n}, v_h\right)_{\Omega_{0,h}\cup\Omega_{1,h}} + (1-\theta)F\big(u_h^{n-1}, v_h, t^{n-1}\big). \end{aligned} \tag{A.7}$$

Appendix B. Weak implementation of the pseudo-periodic BCs

The pseudo-periodic BCs (Eq. (4)) can be implemented weakly through the use of an artificial permeability coefficient, κ^e [15,16]. The artificial permeability condition at the external boundaries take two equations for the master side and the slave side of the mesh. For the master side, it has the following form

$$\boldsymbol{D}_m\nabla U_m\cdot\boldsymbol{n}_m = \kappa^e\big(U_s\, e^{i\,\theta_{ms}} - U_m\big), \tag{B.1}$$

and for the slave-side it has the following form

$$\boldsymbol{D}_s\nabla u_s\cdot\boldsymbol{n}_s = \kappa^e\big(U_m\, e^{i\,\theta_{sm}} - U_s\big), \tag{B.2}$$

where $U_s = U(\boldsymbol{x}_s), U_m = U(\boldsymbol{x}_m), \theta_{ms} = -\theta_{sm} = \gamma\,\boldsymbol{g}\cdot(\boldsymbol{x}_s - \boldsymbol{x}_m)\,\mathcal{F}(t)$. When the master side is considered (Eq. B.1), $\boldsymbol{x}_m$ is the mesh point and $\boldsymbol{x}_s$ is the projection of $\boldsymbol{x}_m$ onto the slave side. Similarly, when the slave side is considered (Eq. B.2), $\boldsymbol{x}_s$ is the mesh point and $\boldsymbol{x}_m$ is the projection of $\boldsymbol{x}_s$ onto the master side. So, the points always align each other but they do not need to be the mesh grid at the same time. So, this method allows for non-matching meshes.

The artificial permeability coefficient κ^e can be chosen to be consistent with the Nitsche's method for the Dirichlet BCs [49] (see also a review in [50] and references therein), i.e $\kappa^e = \max\{\frac{D}{h}\}$ where h is the element size.

To overcome the CFL constraints, the following operator splitting can be used to have an unconditionally stable scheme

$$\begin{aligned} &\boldsymbol{D}_m\nabla U_m\cdot\boldsymbol{n}_m \approx \kappa^e\Big(U_s^{n-1}\, e^{i\,\theta_{ms}^n} - U_m^n\Big), \\ &\boldsymbol{D}_s\nabla U_s\cdot\boldsymbol{n}_s \approx \kappa^e\Big(U_m^{n-1}\, e^{i\,\theta_{sm}^n} - U_s^n\Big), \end{aligned} \tag{B.3}$$

where U^n and U^{n-1} are the approximations at the current and previous time step respectively.

Without imposing the weak pseudo-periodic, the PUFEM with the time-stepping θ-method is stated as: Find $U^n = (U_0^n, U_1^n) \in \boldsymbol{V}^h$ such that

$$\left(\frac{U^n - U^{n-1}}{k^n}, v^h\right)_{\Omega_0\cup\Omega_1} = \theta F(U^n, v^h, t^n) + (1-\theta)F(U^{n-1}, v^h, t^{n-1}), \tag{B.4}$$

for all $v^h = (v_0^h, v_1^h) \in \boldsymbol{V}^h$, where

$$\begin{aligned} F(U,v,t) = &\left(-i\gamma f(t)\boldsymbol{g}\cdot\boldsymbol{x}U - \frac{U}{T_2}, v\right)_{\Omega_0\cup\Omega_1} \\ &-(\boldsymbol{D}\nabla U, \nabla v)_{\Omega_0\cup\Omega_1} - \kappa\langle[\![U]\!], [\![v]\!]\rangle, \end{aligned} \tag{B.5}$$

and $(a,b)_{\Omega_0^h\cup\Omega_1^h} = \left((1-\Phi^h)a_0, b_0\right)_{\Omega^h} + \left(\Phi^h a_1, b_1\right)_{\Omega^h}, \Phi^h$ is an element-wise constant function.

The bilinear and linear forms are defined by

$$\begin{aligned}\boldsymbol{a}(U^n, v^h) &= \left(\tfrac{U^n}{k^n}, v^h\right)_{\Omega_0\cup\Omega_1} - \theta F(U^n, v^h, t^n),\\ \boldsymbol{L}(v^h) &= \left(\tfrac{U^{n-1}}{k^n}, v^h\right)_{\Omega_0\cup\Omega_1} + (1-\theta)F\left(U^{n-1}, v^h, t^{n-1}\right).\end{aligned} \tag{B.6}$$

The linear system of equations corresponding to the bilinear and linear forms (Eq. (B.6)) is

$$\boldsymbol{A}\boldsymbol{U}^n = \boldsymbol{F}, \tag{B.7}$$

where

$$\boldsymbol{A} = \boldsymbol{M}(k^n)^{-1} - \theta\left(-\left(i\gamma f^n + \frac{1}{T_2}\right)\boldsymbol{J} - \boldsymbol{S} - \boldsymbol{I}\right). \tag{B.8}$$

Here $\boldsymbol{M}$ and $\boldsymbol{S}$ are referred to as the mass and stiffness matrices respectively, $\boldsymbol{J}$ and $\boldsymbol{I}$ are corresponding to the first and third terms on the right-hand side of F (Eq. (B.5)), i.e $(\boldsymbol{g}\cdot\boldsymbol{x}U, v)$ and $\kappa\langle[\![U]\!],[\![v]\!]\rangle$.

To impose the weak periodic BCs, we plug Eq. (B.3) to the linear and bilinear forms

$$\boldsymbol{a}^*(U_h^n, v_h) = a(U_h^n, v_h) + \theta\kappa^e\left(U_h^n, v_{h\Gamma_m^0\cup\Gamma_m^1} + \langle U_h^n, v_h\rangle_{\Gamma_s^0\cup\Gamma_s^1}\right),$$

$$\boldsymbol{L}^*(v_h) = L(v_h) + (1-\theta)\kappa^e\left(U_{s,h}^{n-1}\, e^{i\theta_{ms}^n}, v_{h\Gamma_m^0\cup\Gamma_m^1} + U_{m,h}^{n-1}\, e^{i\theta_{sm}^n}, v_{h\Gamma_s^0\cup\Gamma_s^1}\right).$$

References

[1] B.D. Hughes, Random Walks and Random Environments, Clarendon Press, Oxford; New York, 1995.

[2] C.-H. Yeh, B. Schmitt, D. Le Bihan, J.-R. Li-Schlittgen, C.-P. Lin, C. Poupon, Diffusion microscopist simulator: a general monte carlo simulation system for diffusion magnetic resonance imaging, PLoS One 8 (8) (2013) e76626, https://doi.org/10.1371/journal.pone.0076626, pONE-D-13-18755[PII]. URL http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3794953/.

[3] M.G. Hall, D.C. Alexander, Convergence and parameter choice for monte-carlo simulations of diffusion mri, IEEE Trans. Med. Imag. 28 (9) (2009) 1354–1364, https://doi.org/10.1109/TMI.2009.2015756.

[4] M. Palombo, C. Ligneul, C. Najac, J. Le Douce, J. Flament, C. Escartin, P. Hantraye, E. Brouillet, G. Bonvento, J. Valette, New paradigm to assess brain cell morphology by diffusion-weighted mr spectroscopy in vivo, Proc. Natl. Acad. Sci. USA 113 (24) (2016) 6671–6676, https://doi.org/10.1073/pnas.1504327113, 201504327[PII] http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4914152/.

[5] K.V. Nguyen, E.H. Garzon, J. Valette, Efficient gpu-based monte-carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy, J. Magn. Reson. https://doi.org/10.1016/j.jmr.2018.09.013. <http://www.sciencedirect.com/science/article/pii/S1090780718302386>.

[6] P.A. Cook, Y. Bai, S. Nedjati-Gilani, K.K. Seunarine, M.G. Hall, G.J.M. Parker, D.C. Alexander, Camino: Open-source diffusion-mri reconstruction and processing, 2006.

[7] S.N. Hwang, C.-L. Chin, F.W. Wehrli, D.B. Hackney, An image-based finite difference model for simulating restricted diffusion, Magn. Reson. Med. 50 (2) (2003) 373–382, https://doi.org/10.1002/mrm.10536.

[8] J. Xu, M. Does, J. Gore, Numerical study of water diffusion in biological tissues using an improved finite difference method, Physics in Medicine and Biology 52 (7). <http://view.ncbi.nlm.nih.gov/pubmed/17374905>.

[9] K.D. Harkins, J.-P. Galons, T.W. Secomb, T.P. Trouard, Assessment of the effects of cellular tissue properties on ADC measurements by numerical simulation of water diffusion, Magn. Reson. Med. 62 (6) (2009) 1414–1422, https://doi.org/10.1002/mrm.22155.

[10] G. Russell, K.D. Harkins, T.W. Secomb, J.-P. Galons, T.P. Trouard, A finite difference method with periodic boundary conditions for simulations of diffusion-weighted magnetic resonance experiments in tissue, Phys. Med. Biol. 57 (4) (2012) N35. http://stacks.iop.org/0031-9155/57/i=4/a=N35.

[11] B.F. Moroney, T. Stait-Gardner, B. Ghadirian, N.N. Yadav, W.S. Price, Numerical analysis of NMR diffusion measurements in the short gradient pulse limit, J. Magn. Reson. 234 (2013) 165–175. http://www.sciencedirect.com/science/article/pii/S1090780713001572.

[12] D.V. Nguyen, J.-R. Li, D. Grebenkov, D.L. Bihan, A finite elements method to solve the blochtorrey equation applied to diffusion magnetic resonance imaging, J. Comput. Phys. 263 (Supplement C) (2014) 283–302, https://doi.org/10.1016/j.jcp.2014.01.009, URL http://www.sciencedirect.com/science/article/pii/S0021999114000308.

[13] L. Beltrachini, Z.A. Taylor, A.F. Frangi, A parametric finite element solution of the generalised blochtorrey equation for arbitrary domains, J. Magn. Reson. 259 (Supplement C) (2015) 126–134, https://doi.org/10.1016/j.jmr.2015.08.008, URL http://www.sciencedirect.com/science/article/pii/S1090780715001743.

[14] D.V. Nguyen, J.R. Li, D.S. Grebenkov, D.L. Bihan, Modeling the diffusion magnetic resonance imaging signal inside neurons, J. Phys: Conf. Ser. 490 (1) (2014), 012013 http://stacks.iop.org/1742-6596/490/i=1/a=012013.

[15] V.D. Nguyen, A FEniCS-HPC framework for multi-compartment Bloch-Torrey models, Vol. 1, 2016, pp. 105–119, QC 20170509. <https://www.eccomas2016.org/>.

[16] V.-D. Nguyen, J. Jansson, J. Hoffman, J.-R. Li, A partition of unity finite element method for computational diffusion mri, J. Comput. Phys. 375 (2018) 271–290, https://doi.org/10.1016/j.jcp.2018.08.039. http://www.sciencedirect.com/science/article/pii/S0021999118305709.

[17] V.-D. Nguyen, J. Jansson, H.T.A. Tran, J. Hoffman, J.-R. Li, Diffusion mri simulation in thin-layer and thin-tube media using a discretization on manifolds, J. Magn. Reson. 299 (2019) 176–187, https://doi.org/10.1016/j.jmr.2019.01.002. http://www.sciencedirect.com/science/article/pii/S1090780719300023.

[18] J.-R. Li, V.-D. Nguyen, T.N. Tran, J. Valdman, C.-B. Trang, K.V. Nguyen, D.T.S. Vu, H.A. Tran, H.T.A. Tran, T.M.P. Nguyen, Spindoctor: a matlab toolbox for diffusion mri simulation, NeuroImage 202 (2019), https://doi.org/10.1016/j.neuroimage.2019.116120, 116120 http://www.sciencedirect.com/science/article/pii/S1053811919307116.

[19] A. Logg, K.-A. Mardal, G.N. Wells, Automated solution of differential equations by the finite element method: the FEniCS book, Springer Verlag, 2012, xIII, 723 s.: ill.

[20] FEniCS, Fenics project, <http://www.fenicsproject.org>.

[21] J.S. Hale, L. Li, C.N. Richardson, G.N. Wells, Containers for portable, productive, and performant scientific computing, Comput. Sci. Eng. 19 (6) (2017) 40–50, https://doi.org/10.1109/MCSE.2017.2421459.

[22] H.C. Torrey, Bloch equations with diffusion terms, Phys. Rev. 104 (1956) 563–565, https://doi.org/10.1103/PhysRev. 104.563. https://link.aps.org/doi/10.1103/PhysRev.104.563.

[23] J.E. Tanner, Transient diffusion in a system partitioned by permeable barriers. application to nmr measurements with a pulsed field gradient, J. Chem. Phys. 69 (4) (1978) 1748–1754, https://doi.org/10.1063/1.436751.

[24] E.O. Stejskal, J.E. Tanner, Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient, J. Chem. Phys. 42 (1) (1965) 288–292, https://doi.org/10.1063/1.1695690.

[25] M.D. Does, E.C. Parsons, J.C. Gore, Oscillating gradient measurements of water diffusion in normal and globally ischemic rat brain, Magn. Reson. Med. 49 (2) (2003) 206–215, https://doi.org/10.1002/mrm.10385.

[26] N. Shemesh, S.N. Jespersen, D.C. Alexander, Y. Cohen, I. Drobnjak, T.B. Dyrby, J. Finsterbusch, M.A. Koch, T. Kuder, F. Laun, M. Lawrenz, H. Lundell, P.P. Mitra, M. Nilsson, E. Zarslan, D. Topgaard, C.-F. Westin, Conventions and nomenclature for double diffusion encoding nmr and mri, Magn. Reson. Med. 75 (1) (2016) 82–87, https://doi.org/10.1002/mrm.25901. https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.25901.

[27] B. Dhital, M. Reisert, E. Kellner, V.G. Kiselev, Intra-axonal diffusivity in brain white matter, NeuroImage 189 (2019) 543–550, https://doi.org/10.1016/j.neuroimage.2019.01.015. http://www.sciencedirect.com/science/article/pii/S1053811919300151.

[28] D.S. Novikov, E. Fieremans, S.N. Jespersen, V.G. Kiselev, Quantifying brain microstructure with diffusion MRI: theory and parameter estimation, NMR Biomed. 32 (4) (2019), https://doi.org/10.1002/nbm.3998, e3998 https://onlinelibrary.wiley.com/doi/abs/10.1002/nbm.3998.

[29] R.N. Henriques, S.N. Jespersen, N. Shemesh, Microscopic anisotropy misestimation in spherical-mean single diffusion encoding mri, Magn. Reson. Med. 81 (5) (2019) 3245–3261, https://doi.org/10.1002/mrm.27606. https://onlinelibrary.wiley.com/doi/pdf/10.1002/mrm.27606.

[30] D. Topgaard, Multidimensional diffusion mri, J. Magn. Reson. 275 (2017) 98–113, https://doi.org/10.1016/j.jmr.2016.12.007. http://www.sciencedirect.com/science/article/pii/S1090780716302701.

[31] Z. Yuan, J. Fish, Toward realization of computational homogenization in practice, Int. J. Numer. Meth. Eng. 73 (2008) 361–380, https://doi.org/10.1002/nme.2074.

[32] B. Bashari Rad, H. Bhatti, M. Ahmadi, An introduction to docker and analysis of its performance, IJCSNS Int. J. Comput. Sci. Network Secur. 173 (2017) 8.

[33] G.M. Kurtzer, V. Sochat, M.W. Bauer, Singularity: scientific containers for mobility of compute, PLOS ONE 12 (5) (2017) 1–20, https://doi.org/10.1371/journal.pone.0177459.

[34] E. National, Academies of Sciences, Medicine, Reproducibility and Replicability in Science, The National Academies Press, Washington, DC, 2019, doi: 10.17226/25303.

[35] J. Melenk, I. Babuka, The partition of unity finite element method: basic theory and applications, Comput. Methods Appl. Mech. Eng. 139 (1) (1996) 289–314, https://doi.org/10.1016/S0045-7825(96)01087-0. http://www.sciencedirect.com/science/article/pii/S0045782596010870.

[36] B. Kehlet, A. Logg, J. Ring, G.N. Wells, FEniCS project, https://bitbucket.org/fenics-project/mshr/, 2019.

[37] A. Logg, G.N. Wells, Dolfin: automated finite element computing, ACM Trans. Math. Softw. 37 (2) (2010) 20:1–20:28, https://doi.org/10.1145/1731022.1731030.

[38] C. Geuzaine, J.F. Remacle, Gmsh: a three-dimensional finite element mesh generator with built-in pre- and post-processing facilities, Int. J. Numer. Methods Eng.
[39] A. Ribes, C. Caremoli, Salom platform component model for numerical simulation, in: 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), Vol. 2, 2007, pp. 553–564. https://doi.org/10.1109/COMPSAC.2007.185.
[40] Beta cae systems, ansa pre-processor: The advanced cae pre-processing software for complete model build up., <https://www.beta-cae.com>.
[41] N. Schlömer, meshio, <https://github.com/nschloe/meshio>, 2019.
[42] G. Inc., Google colaboratory, <https://github.com/jupyter/colaboratory>, 2014.
[43] J.H. et al., Matplotlib.
[44] K.I. Sandia Corporation, Paraview.
[45] D.S. Grebenkov, Pulsed-gradient spin-echo monitoring of restricted diffusion in multilayered structures, J. Magn. Reson. 205 (2) (2010) 181–195, https://doi.org/10.1016/j.jmr.2010.04.017. http://www.sciencedirect.com/science/article/pii/S1090780710001199.
[46] J.P. Kinney, J. Spacek, T.M. Bartol, C.L. Bajaj, K.M. Harris, T.J. Sejnowski, Extracellular sheets and tunnels modulate glutamate diffusion in hippocampal neuropil, J. Comparat. Neurol. 521 (2) (2013) 448–464, https://doi.org/10.1002/cne.23181. https://onlinelibrary.wiley.com/doi/pdf/10.1002/cne.23181.
[47] L. Carim-Todd, K.G. Bath, G. Fulgenzi, S. Yanpallewar, D. Jing, C.A. Barrick, J. Becker, H. Buckley, S.G. Dorsey, F.S. Lee, L. Tessarollo, Endogenous truncated trkb.t1 receptor regulates neuronal complexity and trkb kinase receptor function in vivo, J. Neurosci. 29 (3) (2009) 678–685, https://doi.org/10.1523/JNEUROSCI.5060-08.2009. http://www.jneurosci.org/content/29/3/678.full.pdf.
[48] S. Developers, Scipy, <https://www.scipy.org>, 2001.
[49] J. Nitsche, Über ein variationsprinzip zur lösung von dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind, Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg 36 (1) (1971) 9–15, https://doi.org/10.1007/BF02995904.
[50] P. Hansbo, Nitsche's method for interface problems in computational mechanics, GAMM-Mitteilungen 28 (2) (2005) 183–206, https://doi.org/10.1002/gamm.201490018.

Time-resolved Adaptive Direct FEM Simulation of High-lift Aircraft Configurations

Johan Jansson, Ezhilmathi Krishnasamy, Massimiliano Leoni, Niclas Jansson and Johan Hoffman

Abstract We present an adaptive finite element method for time-resolved simulation of aerodynamics without any turbulence model parameters, which is applied to a benchmark problem from the HiLiftPW-3 workshop to compute the flow past a JAXA Standard Model (JSM) aircraft model at realistic Reynolds number. The mesh is automatically constructed by the method as part of an adaptive algorithm based on a posteriori error estimation using adjoint techniques. No explicit turbulence model is used, and the effect of unresolved turbulent boundary layers is modeled by a simple parametrization of the wall shear stress in terms of a skin friction. In the case of very high Reynolds numbers we approximate the small skin friction by zero skin friction, corresponding to a free slip boundary condition, which results in a computational model without any model parameter to be tuned, and without the need for costly boundary layer resolution. We introduce a numerical tripping noise term to act as a seed for growth of perturbations, the results support that this triggers the correct physical separation at stall, and has no significant effect pre-stall. We show that the methodology quantitavely and qualitatively captures the main features of the JSM experiment - aerodynamic forces and the stall mechanism - with a much coarser mesh resolution and lower computational cost than the state of the art methods in the field, with convergence under mesh refinement by the adaptive method. Thus, the simulation methodology appears to be a possible answer to the challenge of reliably predicting turbulent-separated flows for a complete air vehicle.

Johan Jansson (Assistant Professor, e-mail: jjan@kth.se),
Ezhilmathi Krishnasamy (PhD candidate, e-mail: ekrishnasamy@bcamath.org), and
Massimiliano Leoni (PhD candidate, e-mail: mleoni@kth.se),
Computational Science and Technology, CSC, KTH, SE-10044 Stockholm, Sweden and
BCAM - Basque Center for Applied Mathematics, Bilbao, Spain

Niclas Jansson (Postdoctoral Researcher, e-mail: njansson@kth.se) and
Johan Hoffman (Professor, e-mail: jhoffman@kth.se),
Computational Science and Technology, CSC, KTH, SE-10044 Stockholm, Sweden

Nomenclature

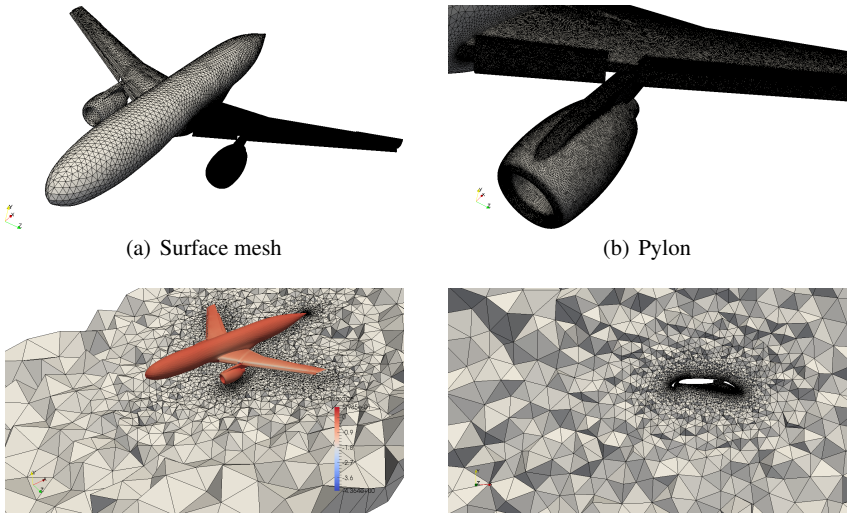
cl	lift coefficient (dimensionless)
cd	drag coefficient (dimensionless)
cp	pressure coefficient (dimensionless)
h	diameter of tetrahedron in finite element mesh (m)
k	time step (s)
$\mathbf{n}$	normal unit vector (dimensionless)
P	computed pressure (Pa)
p	pressure (Pa)
q	pressure test function (Pa)
Re	Reynolds number (dimensionless)
t	time variable (s)
T	end time (s)
$\mathbf{U}$	computed velocity ($\mathrm{m\,s^{-1}}$)
$\mathbf{u}$	velocity ($\mathrm{m\,s^{-1}}$)
$\mathbf{v}$	velocity test function ($\mathrm{m\,s^{-1}}$)
$\mathbf{x}$	space variable (m)
α	angle of attack (dimensionless)
β	friction parameter ($\mathrm{kg\,m^{-2}\,s^{-1}}$)
ν	Kinematic viscosity ($\mathrm{m^2\,s^{-1}}$)
τ	tangent unit vector (dimensionless)

1 Introduction

The main challenge today in Computational Fluid Dynamics (CFD) for aerodynamics is to reliably predict turbulent-separated flows Witherden & Jameson (2017), Slotnick et al. (2014), specifically for a complete air vehicle. This is our focus in this paper.

We present an adaptive finite element method without turbulence modeling parameters for time-resolved simulation of aerodynamics, together with results stemming from the 3rd AIAA CFD High-Lift Prediction Workshop (HiLiftPW-3) which was held in Denver, Colorado, on June 3rd–4th 2017. The benchmark was a high-lift configuration of the JSM aircraft model shown in Figure 1 at a Reynolds number realistic for flight conditions. The purpose of the workshop is to assess the capability of state of the art CFD codes and methods.

Turbulent flows present features on a range of scales, from the scale of the aircraft down to the Kolmogorov dissipation scale. Direct numerical simulation (DNS) is not feasible for a full aircraft at realistic Reynolds numbers, instead the Reynolds Averaged Navier-Stokes equations (RANS) have long been the state of the art in industry Shan et al. (2005). RANS methods do not provide a full resolution of the flow field but simulate the mean field and introduce turbulence models to make it



(a) Surface mesh (b) Pylon

(c) Volume mesh (d) Wing slice

Fig. 1 Overview of the JSM aircraft model and starting mesh for the adaptive method.

up for the unresolved dynamics. In particular, standard RANS do not resolve the transient flow field, but a statistical average of the turbulent flow.

In contrast, Large Eddy Simulations (LES) Sagaut (2005) resolve the dynamics of a filtered flow field, at the cost of higher mesh resolution than RANS, with subgrid models for unresolved scales. Both RANS and LES, and hybrids such as DES, introduce model parameters that need to be tuned to the problem at hand, and the results are highly sensitive to the design of the computational mesh Moin & You (2008), Huang et al. (2004), Spalart (2009), Piomelli & Balaras (2002), Mellen et al. (2003). In particular, turbulent boundary layers cannot be resolved and must be modelled. Boundary layer models require tailored boundary layer meshes, which are expensive in terms of both mesh density and manual work. Witherden and Jameson in Witherden & Jameson (2017) state that "as a community we are still far away from LES of a complete air vehicle".

The method we present is an adaptive finite element method without explicit turbulence model and boundary layer model, thus without model parameters and without the need for a boundary layer mesh. The mesh is automatically constructed by the method as part of the computation through an adaptive procedure based on a posteriori error estimation using adjoint techniques. Dissipation of turbulent kinetic energy is provided by residual-based numerical stabilization. The method is thus purely based on the Navier-Stokes equations, no other modeling assumptions are made.

We model the effect of turbulent boundary layers by a parametrization of the wall shear stress in terms of a skin friction. For very high Reynolds numbers we

approximate the small skin friction by zero skin friction, corresponding to a free slip boundary condition, which results in a computational method without any model parameters that need to be tuned, and without the need for costly boundary layer resolution.

In this paper we give the main components of the simulation methodology and we present our results stemming from the HiLiftPW-3, where we highlight the non-standard aspects of the methodology and discuss the results in relation to the experiments. The HiLiftPW-3 specified two variants of the JSM, one without pylon (or nacelle) and one with the pylon included in the geometry ("pylon on"). The difference in the aerodynamic forces between the two variants measured in experiments are small, typically less than 2 %. For this reason we will focus only on the "pylon on" variant with the aim of validating our methodology.

The workshop guidelines prescribed the study of these two variants either with a fixed mesh or, more interestingly, using mesh adaptation techniques. Considering the nature of our method, which intimately depends on its adaptive procedure, we concentrated on the latter study. We did not use the provided computational meshes, but instead generated more suitable ones for our methodology starting from the provided CAD files. We would like to point out that our adaptive methodology does not require any ad-hoc meshing procedure aimed at helping the solver identify flow features that are qualitatively known before starting the computations. Not only does this simplify the meshing procedure, which can now be carried out by non-specialized software (and scientists), but it also makes it faster: the only thing that we need is an initial mesh that captures the geometry of the object; this is due to the fact that the generated mesh loses memory of the underlying CAD model, and therefore the refinement of boundary triangles cannot correct a rough initial approximation of the CAD geometry. We plan to get rid of this constraint in the near future, implementing the functionality to refine boundary cells with the new vertices projected on the CAD model. Once we have a sufficiently accurate surface description, however, we can let the mesh be coarse in the volume part, which will be refined iteratively by the adaptive algorithm.

This convenient approach allows us to perform computations starting with rather coarse meshes, increasing the number of cells only where needed in order to best utilize the available computational resources. Our initial mesh for the JSM case have about 25M cells.

We find that the simulation results compare very well with experimental data for all the angles of attack that we studied; moreover, we show mesh-convergence by the adaptive method, while using a relatively low number of spatial degrees of freedom. The low computational cost also allows for a time-resolved simulation, which provides additional results that cannot be obtained from a stationary simulation, such as the ones based on Reynolds-averaged Navier-Stokes equations (RANS).

Thus, the simulation methodology appears to be a possible answer to the challenge of reliably predicting turbulent-separated flows for a complete air vehicle. We specifically here present simulation results reproducing the physically correct stall mechanism of large-scale separation at the wing-body juncture, which is promising for our continuing work on validating the methodology.

2 Simulation Methodology

In contrast to the statistical averages of RANS and the filtered solutions of LES, our simulation method is based on computational approximation of weak solutions to the Navier-Stokes equations (NSE), that satisfy the NSE in variational form integrated against a class of test functions.

Finite element methods (FEM) are based on a variational form of the NSE, and if the method satisfies certain conditions on stability and consistency, the FEM solutions converge towards a weak solution to the NSE as the finite element mesh is refined Hoffman & Johnson (2007). We refer to such FEM as a General Galerkin (G2) method, or a Direct Finite Element simulation (DFS).

The resolution in DFS is set by the mesh size, and no turbulence model is introduced. Dissipation of turbulent kinetic energy in under-resolved parts of the flow is provided by the numerical stabilization of G2 in the form of a weighted least squares method based on the residual of NSE.

The mesh is adaptively constructed based on a posteriori estimation of the error in chosen goal or target functionals, such as drag and lift forces. The a posteriori error estimates take the form of a residual weighted by the solution of an adjoint problem, which is computed separately using a similar stabilized FEM method Hoffman & Johnson (2007). The adaptive algorithm starts from a coarse mesh, which is locally refined each iteration based on the a posteriori error estimates.

We use a free slip boundary condition as a model for high Reynolds number turbulent boundary layers with small skin friction. This means that boundary layers are left unresolved, and that no boundary layer mesh is needed.

This methodology has been validated on a number of standard benchmark problems in the literature Hoffman (2005, 2006), Hoffman & Johnson (2006*b*), Hoffman (2009), including for an aircraft model for the HiLiftPW-2 **?** and we find that also for the benchmark considered in this paper the method is very efficient and provides results close to the experimental reference data.

We have used a low order finite element discretization on unstructured tetrahedral meshes, which we refer to as cG(1)cG(1), id est, continuous piecewise linear approximation in space and time.

2.1 The cG(1)cG(1) method

As the basic model for incompressible Newtonian fluid flow, we consider the NSE with constant kinematic viscosity $\nu > 0$, enclosed in $\Omega \subset \mathbb{R}^3$, with boundary Γ, over a time interval $I = [0,T]$:

$$\begin{cases} \dot{\mathbf{u}} + (\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - 2\nu\nabla\cdot\varepsilon(\mathbf{u}) = f, & (\mathbf{x},t) \in \Omega\times I, \\ \nabla\cdot\mathbf{u} = 0, & (\mathbf{x},t) \in \Omega\times I, \\ \mathbf{u}(\mathbf{x},0) = \mathbf{u}^0(\mathbf{x}), & \mathbf{x}\in\Omega, \end{cases} \tag{1}$$

with $\mathbf{u}(\mathbf{x},t)$ the velocity vector, $p(\mathbf{x},t)$ the pressure, $\mathbf{u}^0(\mathbf{x})$ the initial data and $f(\mathbf{x},t)$ a body force. Moreover, $\sigma_{ij} = 2\nu\varepsilon_{ij}(\mathbf{u}) - p\delta_{ij}$ is the stress tensor, with the strain rate tensor $\varepsilon_{ij}(\mathbf{u}) = 1/2(\partial u_i/\partial x_j + \partial u_j/\partial x_i)$, and δ_{ij} the Kronecker delta. The relative importance of viscous and inertial effects in the flow is determined by the Reynolds number $Re = \mathbf{U}L/\nu$, where $\mathbf{U}$ and L are characteristic velocity and length scales.

The cG(1)cG(1) method is based on the continuous Galerkin method cG(1) in space and time. With cG(1) in time, the trial functions are continuous, piecewise linear and the test functions piecewise constant. cG(1) in space corresponds to both test functions and trial functions being continuous, piecewise linear.

Let $0 = t_0 < t_1 < ... < t_N = T$ be a sequence of discrete time steps, with associated time intervals $I_n = (t_{n-1}, t_n)$ of length $k_n = t_n - t_{n-1}$, and let $W \subset H^1(\Omega)$ be a finite element space consisting of continuous, piecewise linear functions on a tetrahedral mesh $\mathscr{T} = \{K\}$ of mesh size $h(\mathbf{x})$, with $W_{\mathbf{w}}$ the functions $\mathbf{v} \in W$ satisfying the Dirichlet boundary condition $\mathbf{v}|_\Gamma = \mathbf{w}$.

We seek $\hat{\mathbf{U}} = (\mathbf{U}, P)$, continuous piecewise linear in space and time, and the cG(1)cG(1) method for the NSE with homogeneous Dirichlet boundary conditions reads: for $n = 1, ..., N$ find $(\mathbf{U}^n, P^n) \equiv (\mathbf{U}(t_n), P(t_n))$, with $\mathbf{U}^n \in V_0 \equiv [W_0]^3$ and $P^n \in W$, such that:

$$\begin{aligned} &((\mathbf{U}^n - \mathbf{U}^{n-1})k_n^{-1} + \bar{\mathbf{U}}^n \cdot \nabla \bar{\mathbf{U}}^n, \mathbf{v}) + (2\nu\varepsilon(\bar{\mathbf{U}}^n), \varepsilon(\mathbf{v})) - (P^n, \nabla \cdot \mathbf{v}) \\ &+ (\nabla \cdot \bar{\mathbf{U}}^n, q) + SD_\delta^n(\bar{\mathbf{U}}^n, P^n; \mathbf{v}, q) = (f, \mathbf{v}), \quad \forall \hat{\mathbf{v}} = (\mathbf{v}, q) \in V_0 \times W, \end{aligned} \tag{2}$$

where $\bar{\mathbf{U}}^n = \frac{1}{2}(\mathbf{U}^n + \mathbf{U}^{n-1})$ is piecewise constant in time over I_n, with the stabilizing term

$$\begin{aligned} &SD_\delta^n(\bar{\mathbf{U}}^n, P^n; \mathbf{v}, q) \equiv \\ &\quad (\delta_1(\bar{\mathbf{U}}^n \cdot \nabla \bar{\mathbf{U}}^n + \nabla P^n - f), \bar{\mathbf{U}}^n \cdot \nabla \mathbf{v} + \nabla q) + (\delta_1 \nabla \cdot \bar{\mathbf{U}}^n, \nabla \cdot \mathbf{v}), \end{aligned} \tag{3}$$

and

$$\begin{aligned} (\mathbf{v}, \mathbf{w}) &= \sum_{K \in \mathscr{T}} \int_K \mathbf{v} \cdot \mathbf{w}\, dx, \\ (\varepsilon(\mathbf{v}), \varepsilon(\mathbf{w})) &= \sum_{i,j=1}^{3} (\varepsilon_{ij}(\mathbf{v}), \varepsilon_{ij}(\mathbf{w})), \end{aligned}$$

with the stabilization parameter $\delta_1 = \kappa_1 h$, where κ_1 is a positive constant of unit size. We choose a time step size $k_n = C_{CFL} \min_{\mathbf{x} \in \Omega} h/|\mathbf{U}^{n-1}|$, with C_{CFL} typically in the range $[0.5, 20]$. The resulting non-linear algebraic equation system is solved with a robust Schur-type fixed-point iteration method Houzeaux et al. (2009).

2.2 The Adaptive Algorithm

A simple description of the adaptive algorithm, starting from $i = 0$, reads:

1. For the mesh $\mathscr{T}_i$: solve the primal and (linearized) dual problems for the primal solution $(\mathbf{U}, P)$ and the dual solution (Φ, Θ).
2. Compute the quantity $\mathscr{E}_K$ for any cell K of $\mathscr{T}_i$. If $\sum_{K\in\mathscr{T}_i} \mathscr{E}_K < TOL$ then stop, else:
3. Mark 5% of the elements with highest $\mathscr{E}_K$ for refinement.
4. Generate the refined mesh $\mathscr{T}_{i+1}$, and goto 1.

Here, $\mathscr{E}_K$ is the *error indicator* for each cell K, which we describe in Section 2.3. For now, it suffices to say that $\mathscr{E}_K$ is a function of the residual of the NSE and of the solution of a linearized dual problem. The formulation of the dual problem includes the definition of a *target functional* for the refinement, which usually enters the dual equations as a boundary condition or as a volume source term. This functional should be chosen according to the problem we are solving. In other words, one needs to ask the right question in order to obtain the correct answer from the algorithm. In this paper our target functional is chosen to be the mean value in time of the aerodynamic forces.

The dual problem can be written as (see Hoffman & Johnson (2006*a*) for more details):

$$\begin{cases} -\dot{\varphi} - (\mathbf{u}\cdot\nabla)\varphi + \nabla\mathbf{U}^\top\varphi + \nabla\theta - \nu\Delta\varphi = \psi_1 & (\mathbf{x},t) \in \Omega\times I \\ \nabla\cdot\varphi = \psi_2 & (\mathbf{x},t)\in\Omega\times I \\ \varphi = \psi_3 & (\mathbf{x},t)\in\Gamma\times I \\ \varphi(\cdot,T) = \psi_4 & \mathbf{x}\in\Omega, \end{cases} \tag{4}$$

where we find that the structure is similar to the primal NSE equations, except that the adjoint problem is linear, the transport is backward in time, and that we have a reaction term $(\nabla\mathbf{U}^\top\varphi)_j = U_{,j}\cdot\varphi$, not present in the primal NSE.

The only other input required from the user is an initial discretization of the geometry, $\mathscr{T}_0$. Since our method is designed for tetrahedral meshes that do not require any special treatment of the near-wall region (no need for a boundary-layer mesh), the initial mesh can be easily created with any standard mesh generation tool.

2.3 A posteriori error estimate for cG(1)cG(1)

The a posteriori error estimate is based on the following theorem (for a detailed proof, see chapter 30 in Hoffman & Johnson (2007)):

Theorem 1. *If $\hat{\mathbf{U}} = (\mathbf{U}, P)$ solves (2), $\hat{\mathbf{u}} = (\mathbf{u}, p)$ is a weak NSE solution, and $\hat{\varphi} = (\varphi, \theta)$ solves an associated dual problem with data $M(\cdot)$, then we have the*

following a posteriori error estimate for the target functional $M(\hat{\mathbf{U}})$ with respect to the reference functional $M(\hat{\mathbf{u}})$:

$$|M(\hat{\mathbf{u}}) - M(\hat{\mathbf{U}})| \leq \sum_{n=1}^{N} \left[\int_{I_n} \sum_{K \in \mathscr{T}_i} |R_1(\mathbf{U},P)_K| \cdot \omega_1 \, dt \right.$$
$$\left. + \int_{I_n} \sum_{K \in \mathscr{T}_i} |R_2(\mathbf{U})_K| \, \omega_2 \, dt + \int_{I_n} \sum_{K \in \mathscr{T}_i} |SD^n_\delta(\hat{\mathbf{U}}; \hat{\varphi})_K| \, dt \right] =: \sum_{K \in \mathscr{T}_i} \mathscr{E}_K$$

with

$$R_1(\mathbf{U},P) = \dot{\mathbf{U}} + (\mathbf{U} \cdot \nabla)\mathbf{U} + \nabla P - 2\nu \nabla \cdot \varepsilon(\mathbf{u}) - f,$$
$$R_2(\mathbf{U}) = \nabla \cdot \mathbf{U}, \tag{5}$$

where $SD^n_\delta(\cdot;\cdot)_K$ is a local version of the stabilization form (3), and the stability weights are given by

$$\omega_1 = C_1 h_K |\nabla \varphi|_K,$$
$$\omega_2 = C_2 h_K |\nabla \theta|_K,$$

where h_K is the diameter of element K in the mesh $\mathscr{T}_i$, and $C_{1,2}$ represent interpolation constants. Moreover, $|w|_K \equiv (\|w_1\|_K, \|w_2\|_K, \|w_3\|_K)$, with $\|w\|_K = (w,w)_K^{1/2}$, and the dot denotes the scalar product in $\mathbb{R}^3$.

For simplicity, it is here assumed that the time derivatives of the dual variables $\hat{\varphi} = (\varphi, \theta)$ can be bounded by their spatial derivatives. Given Theorem 1, we can understand the adaptive algorithm. As mentioned above, the error indicator, $\mathscr{E}_K$, is a function of the residual of the NSE and the solution of a linearized dual problem (a detailed formulation of the dual problem is given in Chapter 14 in Hoffman & Johnson (2007)). Thus, on a given mesh, we must first solve the NSE to compute the residuals, $R_1(\mathbf{U},P)$ and $R_2(\mathbf{U})$, and then a linearized dual problem to compute the weights multiplying the residuals, ω_1 and ω_2. With that information, we are able to compute $\sum_{K \in \mathscr{T}_i} \mathscr{E}_K$ and check it against the given stop criterion. This procedure of solving the forward and backward problems for the NSE is closely related to an optimization loop and can be understood as the problem of finding the "optimal mesh" for a given geometry and boundary conditions, id est, the mesh with the least possible number of degrees of freedom for computing $M(\hat{\mathbf{u}})$ within a given degree of accuracy.

2.4 The Do-nothing Error Estimate and Indicator

To minimize loss of sharpness, we also investigate an approach where the weak form is used directly in a posteriori error estimates, without integration by parts to the strong form, using the Cauchy-Schwarz inequality and interpolation estimates.

We here refer to this direct form of a posteriori error representation by duality as the "do-nothing" approach.

In terms of the exact adjoint solution $\hat{\varphi}$, the output error with respect to a weak solution $\hat{\mathbf{u}}$ can be represented as

$$|M(\hat{\mathbf{u}}) - M(\hat{\mathbf{U}})| = |(R(\hat{\mathbf{U}}), \hat{\varphi})| = |\sum_{K \in \mathscr{T}_i} (R(\hat{\mathbf{U}}), \hat{\varphi})_K| \tag{6}$$

This error representation involves no approximation or inequalities. We thus refer to the following error indicator based on the representation as the *do-nothing error indicator*:

$$e^K \equiv (R(\hat{\mathbf{U}}), \hat{\varphi})_K \tag{7}$$

A computable estimate and an error indicator are again based on the computed approximation $\hat{\varphi}_h$ of the dual solution:

$$|M(\hat{\mathbf{u}}) - M(\hat{\mathbf{U}})| \approx |(R(\hat{\mathbf{U}}), \hat{\varphi}_h)| \tag{8}$$

$$e_h^K \equiv (R(\hat{\mathbf{U}}), \hat{\varphi}_h)_K \tag{9}$$

where we may lose reliability of the global error estimate by the Galerkin orthogonality property, which states that the $(R(\hat{\mathbf{U}}), \hat{\varphi}_h)$ vanishes for a standard Galerkin finite element method if $\hat{\varphi}_h$ is chosen in the same space as the test functions. Although, in the setting of a stabilised finite element method this may not be the case, see Hoffman et al. (2016).

2.5 Turbulent boundary layers

In our work on high Reynolds number turbulent flows Hoffman & Jansson (2010), Hoffman & Johnson (Published Online First at www.springerlink.com: 10 December 2008), Vilela de Abreu et al. (2014) we have chosen to apply a skin friction stress as wall layer model. That is, we append the NSE with the following boundary conditions:

$$\mathbf{u} \cdot \mathbf{n} = 0, \tag{10}$$

$$\beta \mathbf{u} \cdot \tau_k + \mathbf{n}^T \sigma \tau_k = 0, \quad k = 1, 2, \tag{11}$$

for $(\mathbf{x}, t) \in \Gamma_{solid} \times I$, with $\mathbf{n} = \mathbf{n}(\mathbf{x})$ an outward unit normal vector, and $\tau_k = \tau_k(\mathbf{x})$ orthogonal unit tangent vectors of the solid boundary Γ_{solid}. We use matrix notation with all vectors $\mathbf{v}$ being column vectors and the corresponding row vector being denoted by $\mathbf{v}^T$.

With skin friction boundary conditions, the rate of kinetic energy dissipation in cG(1)cG(1) has a contribution of the form

$$\sum_{k=1}^{2} \int_0^T \int_{\Gamma_{solid}} |\beta^{1/2} \bar{\mathbf{U}} \cdot \tau_k|^2 \, ds \, dt, \tag{12}$$

from the kinetic energy which is dissipated as friction in the boundary layer. For high Re, we model $Re \to \infty$ by $\beta \to 0$, so that the dissipative effect of the boundary layer vanishes with large Re. In particular, we have found that a small β does not influence the solution Hoffman & Jansson (2010). For the present simulations we used the approximation $\beta = 0$, which can be expected to be a good approximation for real high-lift configurations, where Re is very high.

2.6 Numerical tripping

The simulation setting so far is idealized in the sense that the inflow is noise-free, the surfaces have no roughness, there are no vibrations in the surface, etc. This is not a realistic setting.

In the DNS community the effect of introducing noise has been investigated in Schlatter & Orlu (2012), and it turns out that in idealized settings different DNS methods and frameworks may get different results for the same problem, but introducing a noise term has the effect of making the results more uniform.

We explore a similar idea here, where we add a volume force term of simple white in a domain approximately the bounding box of the aircraft geometry. We want the noise to only slightly perturb the solution, to act as a seed for growth of perturbations in unstable mechanisms (such as stall), but we do not want the noise to dominate the solution.

To achieve this balanced effect we scale the white noise force term by 5% of the maximum pressure gradient $|\nabla p|$.

We investigate the effect of such numerical tripping in the results section, comparing simulations with and without the tripping. We will see that especially for stall this appears to have a key effect in triggering the correct physical separation.

2.7 The FEniCS-HPC finite element computational framework

The simulations in this article have been computed using the Unicorn solver in the FEniCS-HPC automated FEM software framework.

FEniCS-HPC Hoffman et al. (2015) is an open source framework for the automated solution of PDEs on massively parallel architectures, providing automated evaluation of variational forms whose description is given in a high-level mathematical notation, duality-based adaptive error control, implicit turbulence modeling by use of stabilized FEM and strong linear scaling up to thousands of cores Hoffman et al. (2013, 2012), Jansson et al. (2012), Kirby (2012), Logg, Ølgaard, Rognes & Wells (2012), Hoffman, Jansson, Jansson & Nazarov (2011), Hoffman,

Jansson, Jansson, Johnson & de Abreu (2011). FEniCS-HPC is a branch of the FEniCS Logg, Mardal, Wells et al. (2012), FEniCS (2003) framework focusing on high performance on massively parallel architectures.

Unicorn is solver technology (models, methods, algorithms and software) with the goal of automated high performance simulation of realistic continuum mechanics applications, such as drag or lift computation for fixed or flexible objects (FSI) in turbulent incompressible or compressible flow. The basis for Unicorn is Unified Continuum (UC) modeling Hoffman, Jansson & Stöckli (2011) formulated in Euler (laboratory) coordinates, together with the General Galerkin (G2) adaptive stabilized finite element discretization described above.

The simulations in this paper were run on supercomputer resources described in the Acknowledgments section, and took ca. 10h on the finest mesh for the whole time interval using ca. 1000 cores.

3 Results

We have performed simulations with the adaptive DFS methodology using the Unicorn/FEniCS-HPC framework for the JSM "pylon on" variant of the HiLiftPW-3 benchmark for the angles 4.36°, 10.58°, 18.58°, 21.57° and 22.58°. All angles except 22.58° have rich experimental data including forces, cp and oil film provided by the workshop, which we will compare against below. The angle 22.58° only has force data. The angles 21.57° and 22.58° exhibit stall in the experiment, e.g. large-scale separation leading to loss of lift force. Capturing stall quantitatively and with the correct stall mechanism is an open problem in aerodynamics, we therefore investigate both the angle 21.57°, which is the highest angle with detailed experimental data, as well as 22.58°,

The experiment is a semispan model at $Re = 193$M. However, "free air" computations were requested, and to avoid possible modeling errors introduced by a symmetry plane we model the entire aircraft. However, we choose the output quantity as drag and lift of the left side of the aircraft only, to save computational resources, where we expect the adaptive method to refine in the right half-volume only when there is a significant error contribution to the drag and lift on the left side.

The initial mesh in the adaptive method has ca. 2.5M vertices, and the mesh is then iteratively refined with 5% of the cells in every iteration until we observe mesh convergence in drag and lift, or as many times as we can afford. The finest adapted meshes in our computations presented here have 5M to 10M vertices.

We solve the time-dependent Navier-Stokes equations (1) with a non-dimensional unit inflow velocity over the time interval $I = [0,10]$. For some of the cases close to stall where we observe a longer startup, we extend the time interval to $I = [0,20]$. To compute the aerodynamic coefficients we take the mean value in the last quarter of the time interval, e.g. $[7.5,10]$ or $[15,20]$, respectively.

We have divided this section into three parts:

1. Detailed comparison of aerodynamic forces against the experiments including convergence of the adaptive method and analysis of stall.
2. Detailed comparison of the pressure coefficients cp against the experimental data, including analysis of cp in the stall regime.
3. Flow visualizations are presented, including dual quantities acting as weights in the error estimates, and comparison of surface velocity against oil film visualizations in the experiment.

3.1 Aerodynamic Forces

The aerodynamic force in the case of zero skin friction is computed as:

$$F = \frac{1}{|I|} \int_I \int_{\Gamma_a} p\mathbf{n}\, ds dt, \tag{13}$$

with Γ_a the left half-boundary of the aircraft. The drag and lift coefficients are then simply the x and y components of F since we have unit inflow.

We use the duality-based "do-nothing" adaptive method, which iteratively refines the mesh by repeatedly solving the primal and dual problem based on the a posteriori error estimate. This generates a sequence of adapted meshes, a procedure that takes the role of the classical *mesh study*.

In Figure 2 we plot the lift coefficient, cl, and drag coefficient, cd, versus the angle of attack, α, for the different meshes from the iterative adaptive method. The size of the dots indicates the iteration number in the adaptive sequence, with larger dots indicating a larger number, that is more refinement. We connect the finest meshes with lines, and also plot the experimental data as lines. For the angles 18.58° and 22.58° we compute the solution both with and without the "numerical tripping" term described in Section 2.6 to assess the dependence on the angle of attack, the tripped cases are plotted in red, and the adaptive sequence shifter somewhat to the right for clarity.

We observe mesh convergence to within 1 % to 2 % for all cases, a close match to the experiments for cl, within circa 5 %, and a small overprediction of circa 10 % for cd, which is consistent with the majority of the participants in HiLiftPW-3 across a range of methods Rumsey (2017), suggesting a systematic error in the problem statement or the experimental data.

For the stall regime angles 18.58°, 21.57° and 22.58° we qualitatively reproduce the stall phenomenon in the experiment – a decrease in cl with increased angle of attack past 21.57°. We observe that the stall angle occurs somewhere between 18.58°, 21.57° which is ca. 1° from the experimental stall angle.

Additionally we verify that the "numerical tripping" functions as expected: the term has no significant impact on the solution for an angle of 18.58°, which is the maximum lift angle and the maximum non-stalling angle, whereas for the stalling angle 22.58° we observe that the tripping has the effect of triggering a large-scale

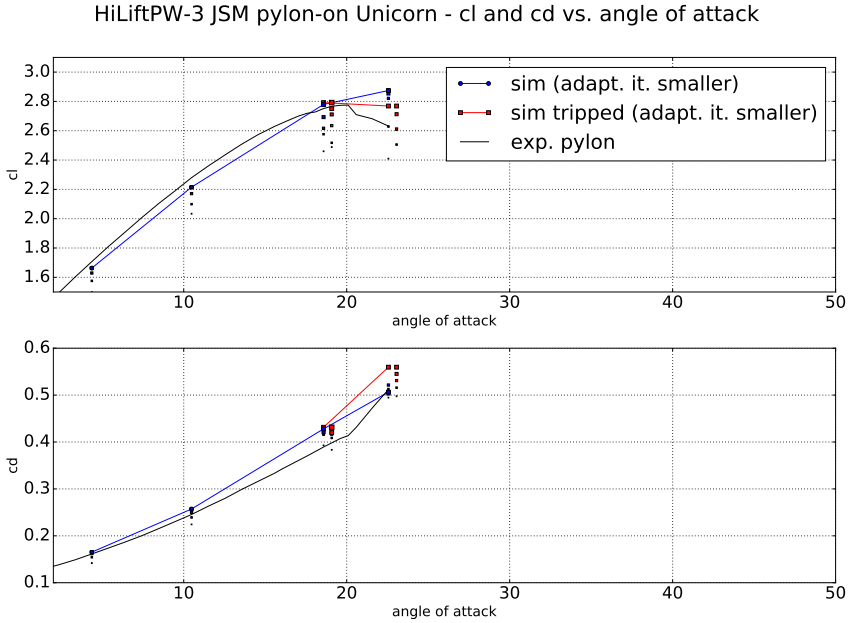


Fig. 2 Lift coefficient, cl, and drag coefficient, cd, versus the angle of attack, α, for the different meshes from the iterative adaptive method.

separation consistent with the stall phenomenon, whereas the untripped case appears to contain too small perturbations for the separation to occur. We analyze the stall mechanism in more detail in the surface velocity visualization below.

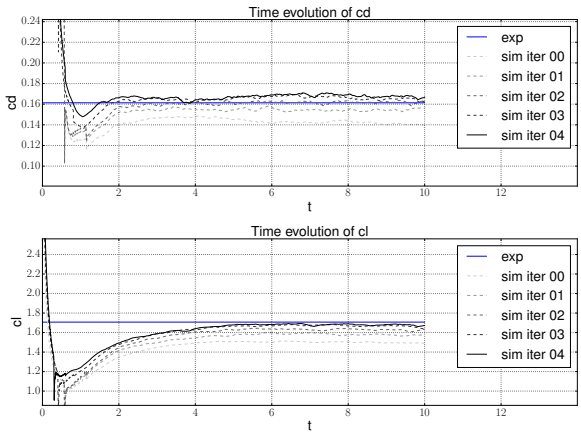
To analyze the variability in time of cd and cl we plot the time evolution for $\alpha = 4.36^\circ$ in Figure 3, untripped with $\alpha = 18.58^\circ$ in Figure 4 and tripped with $\alpha = 18.58^\circ$ in Figure 5.

For the pre-stall cases we observe an initial "startup phase" for $t \in [0,5]$ and then an oscillation around a stable mean value. The effect of the numerical tripping is noise in the cd and cl signals with amplitude of about 1 %.

3.2 Pressure coefficients

The pressure coefficients cp from both simulation on the finest adaptive mesh and experiments are plotted in Figures 7, 8 and 9, for the wing, flap and slat respectively.

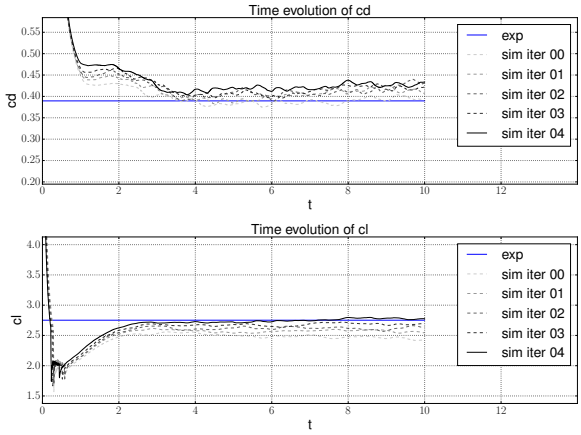
The pressure sensor locations corresponding to the plots are specified in the diagram in Figure 6.



	sim.	exp.	rel. err.
cl	1.680	1.71	1.6%
cd	0.168	0.161	3.9%

(a) $\alpha = 4.36^\circ$

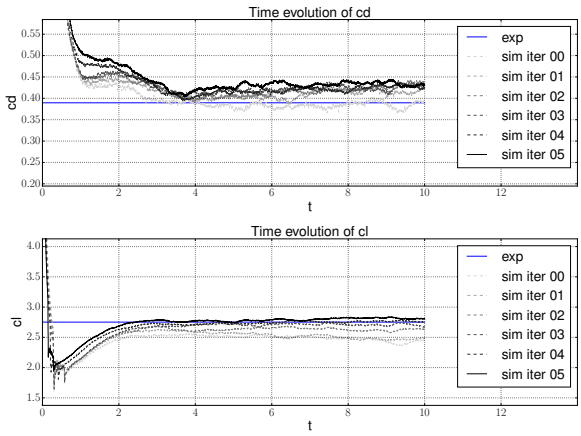
Fig. 3 Time evolution of lift coefficient, *cl*, and drag coefficient, *cd*, and a table of the value for the finest adaptive mesh with relative error compared to the experimental results for $\alpha = 4.36^\circ$.



	sim.	exp.	rel. err.
cl	2.78	2.75	1.0%
cd	0.428	0.390	9.0%

(a) $\alpha = 18.58^\circ$

Fig. 4 Time evolution of lift coefficient, *cl*, and drag coefficient, *cd*, and a table of the value for the finest adaptive mesh with relative error compared to the experimental results for $\alpha = 18.58^\circ$, untripped.



	sim.	exp.	rel. err.
cl	2.80	2.75	1.6%
cd	0.432	0.390	9.8%

(a) $\alpha = 18.58^\circ$

Fig. 5 Time evolution of lift coefficient, *cl*, and drag coefficient, *cd*, and a table of the value for the finest adaptive mesh with relative error compared to the experimental for $\alpha = 18.58^\circ$ with numerical tripping.

Since the aerodynamic force defined in (13) matches the experiment well, and since it consists of integrals of the pressure weighed by the normal vector, the cp values also have to match the experiment on average. However, the cp plots can give insight into local mechanisms such as separation patterns, an important example being the stall mechanism. These local mechanisms are what we will focus on here.

First of all, we see that for the pre-stall angles $\alpha = 10.48^\circ$ and $\alpha = 18.58^\circ$ the simulation and experiment match very well for the wing and slat, and generally well for the flap, aside from local differences. The cp for the simulation is lower on the upper surface for the flap close to the body (the A-A station). Otherwise the curves generally match.

For the stall regime we analyze both 21.57° where experimental cp are available and 22.56° where experimental cp plots are not available. We compare both against the experimental cp plots for 21.57° to have a margin for if we have stall at a higher angle in the simulation. The simulation matches the experiment very well, there is a small discrepancy for the wing close to the body (the A-A station), but considering that this is where the large-scale separation causing the stall is located, the results match acceptably.

The matching cp curves are consistent with matching *cd* and *cl* from the aerodynamic force plots.

We now compare the tripped and untripped simulation with the experiment at 22.56°, as well as 22.56° in Figure 10 for the wing. We clearly see that the untripped simulation for 22.56° grossly misses the cp on the upper surface at station A-A, near the wing-body junction where the large-scale separation mechanism causing stall is located, while the tripped simulation captures the experimental cp curve well, aside from a slightly lower cp near the leading edge. We conclude that the tripping acts

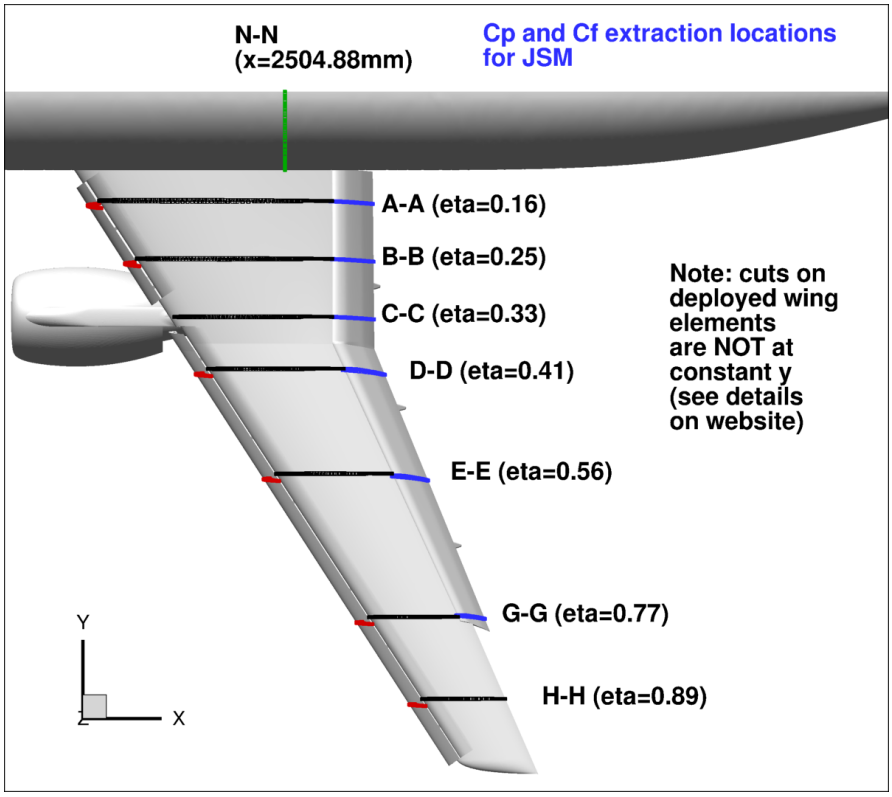


Fig. 6 Diagram of the pressure sensor layout for the JSM configuration showing where the pressure sensors are located and how they are denoted.

to trigger the physically correct separation. At the other stations, D-D and G-G, the tripped and untripped simulations are very similar, indicating that the tripping does not have a significant effect aside from the triggering of the perturbations.

The $\alpha = 21.57^\circ$ simulation is tripped and captures the experiment less well than 22.56°, but better than 22.56° untripped indicating that we may have a ca. 1° later stall angle in the simulation than in the experiment.

3.3 Flow and Adaptive Mesh Refinement Visualization

Here we concentrate on presenting effective visualization of the flow and the adaptive mesh refinement procedure. Our aim is to provide information on the properties and features of the approximated solution and, more importantly, of the approximating procedure, most of which cannot be discerned from one dimensional plots of the

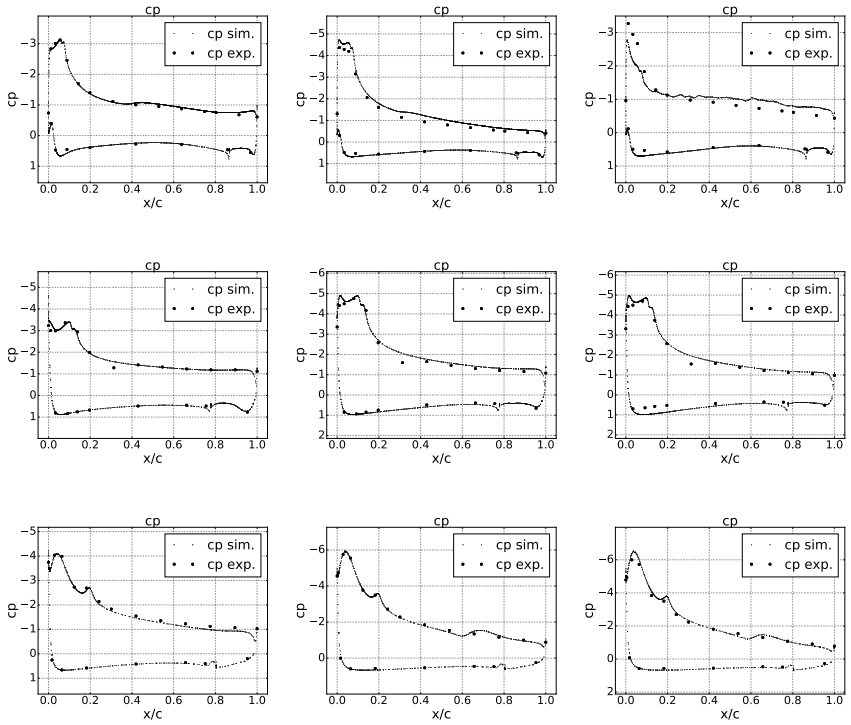


Fig. 7 Pressure coefficients, cp, versus normalized local chord, x/c, for the angles of attack $\alpha = 10.48^\circ$ (left), $\alpha = 18.58^\circ$ (middle) and $\alpha = 22.56^\circ$ (right) at locations A-A (top), D-D (middle) and G-G (bottom) for the wing of JSM pylon on.

pressure coefficient and the aerodynamic forces. Sometimes these more complex visualizations cannot be directly compared to experiments, but still they constitute a qualitative validation of the results.

The first plots that we show are the surface plots of the velocity magnitude on the upper side of the wing. Together with the velocity magnitude surface plots we also report pictures of the oil film experiment that was provided by the organizers as a validation tool. These serve as comparison tools, and we report such comparison in Figure 11.

Some common features intrinsic of the geometry of the JSM aircraft are revealed by the oil film experiment and reproduced by the velocity plots. A pattern of low velocity streaks, alternating with areas of higher velocity, is seen on the suction side of the fixed wing for all angles of attack. This is caused by separation at the slat tracks upstream, which is correctly captured by the numerical solution.

Another characteristic feature of the flow is the turbulent separation near the tip of the wing. This is particularly evident in the case $\alpha = 18.59^\circ$,

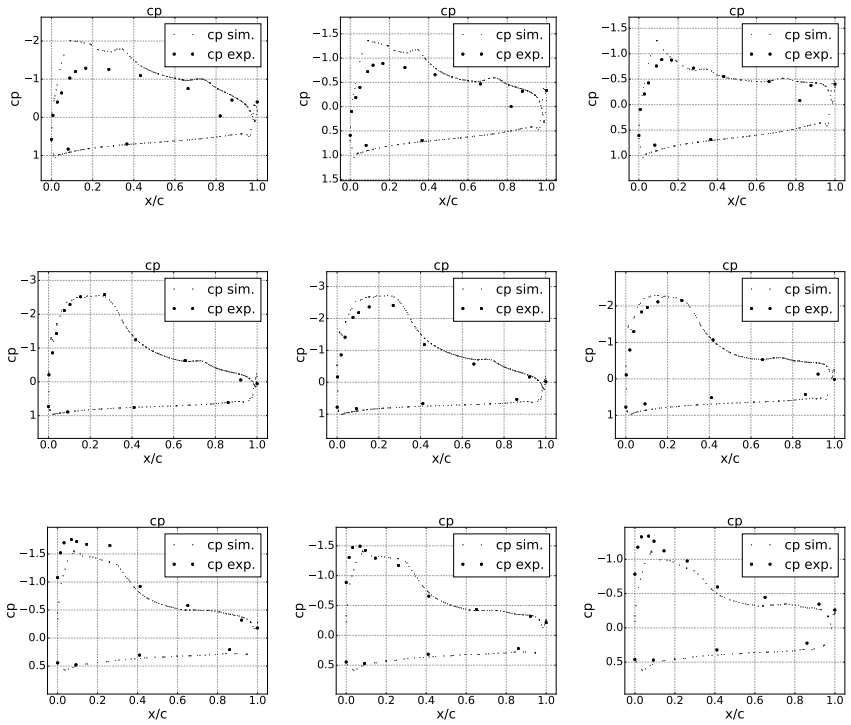


Fig. 8 Pressure coefficients, cp, versus normalized local chord, x/c, for the angles of attack $\alpha =$ 10.48° (left), $\alpha = 18.58^\circ$ (middle) and $\alpha = 22.56^\circ$ (right) at locations A-A (top), D-D (middle) and G-G (bottom) for the flap of JSM pylon on.

Areas that exhibit this kind of flow behavior influence the aerodynamic forces on the aircraft, and indeed in our experimentation we found that computations done on some meshes resulted in wrong predictions of the target functionals, usually yielding lower lift coefficients than the experimental ones. We were able to overcome this intermediate obstacle by refining the surface mesh were the original geometry had a higher curvature. We later interpreted the effectiveness of this workaround as a symptom that the original meshes were unable to capture the surface geometry to a sufficient degree of accuracy, and were for this reason failing at reproducing these complex patterns.

Another interesting visualization technique, which we are about to present, is more closely related to turbulence itself: the Q-criterion Hunt et al. (1988). The Q-criterion was widely used in the literature to visualize turbulent features of fluid flows. The main idea is that it is possible to define a quantity, commonly denoted by the letter Q, whose value is related to the vorticity and thus the visualization of the

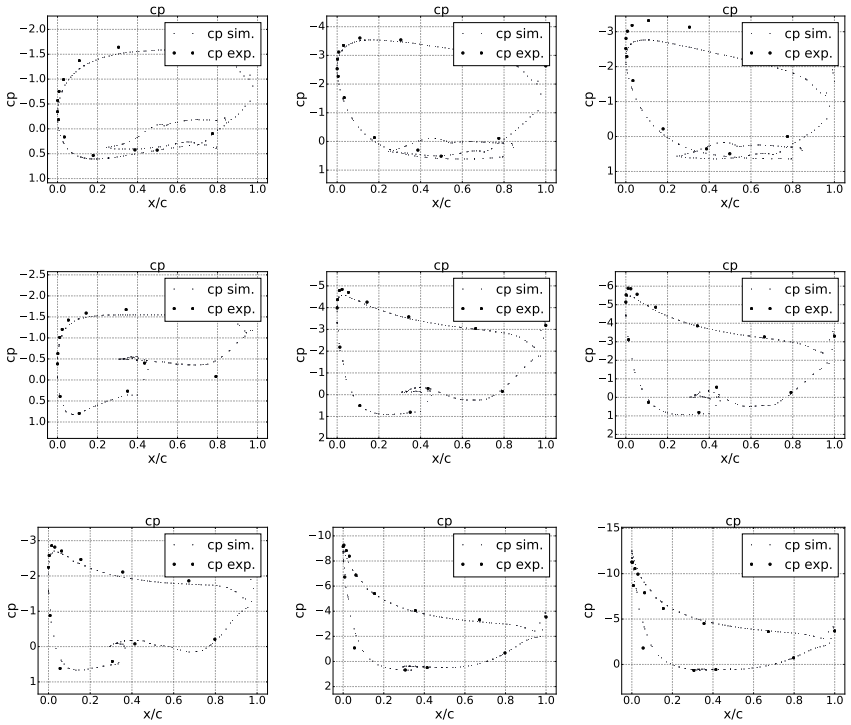


Fig. 9 Pressure coefficients, cp, versus normalized local chord, x/c, in the stall regime for the angles of attack $\alpha = 10.48^\circ$ (left), $\alpha = 18.58^\circ$ (middle) and $\alpha = 22.56^\circ$ (right) at locations A-A (top), D-D (middle) and G-G (bottom) for the slat of JSM pylon on.

isocontours of Q is claimed to give visual information on the presence and location of vortexes within the flow field.

The Q-criterion for the case of the airplane with pylon is displayed in Figure 12 for three different angles of attack.

Once again, the visualization technique highlights the same pattern as in the previous case: the isosurfaces assume a characteristic V shape along the interfaces between the fast and slow velocity regions on the suction side of the wing. Not only that, but we can also clearly distinguish a clustering of these isosurfaces near the tip of the wing, matching the position of the turbulent separation zone that we mentioned above. The Q-criterion visualizations are consistent with the surface velocity plots, and this internal coherence increases our trust in the computational results.

Let us now turn our attention to the adaptive procedure which produces the successive approximations of the fluid flow. As we described above, the mesh refinement solution is driven by the residual of the Navier-Stokes equations and the solu-

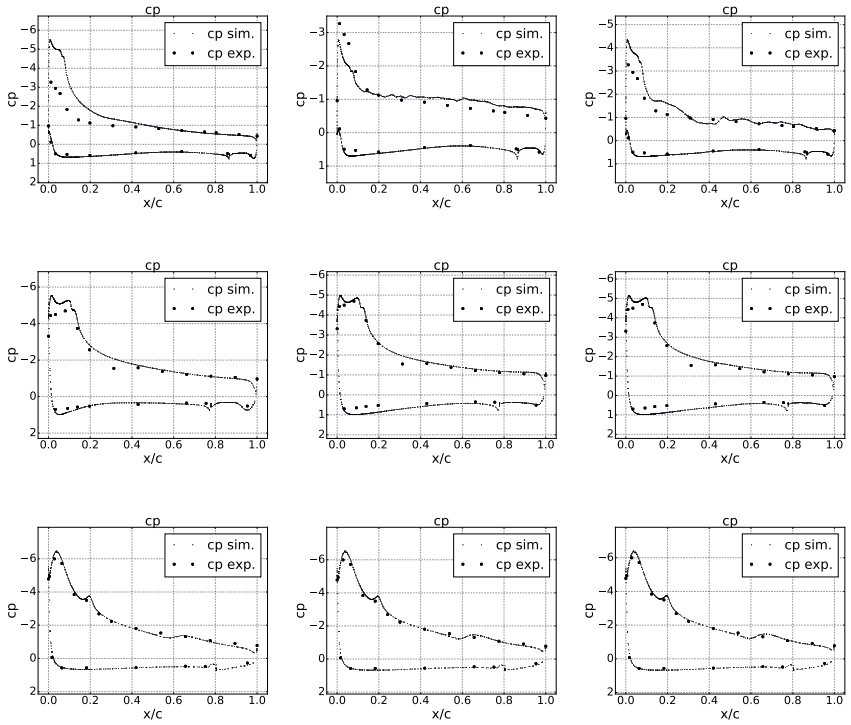
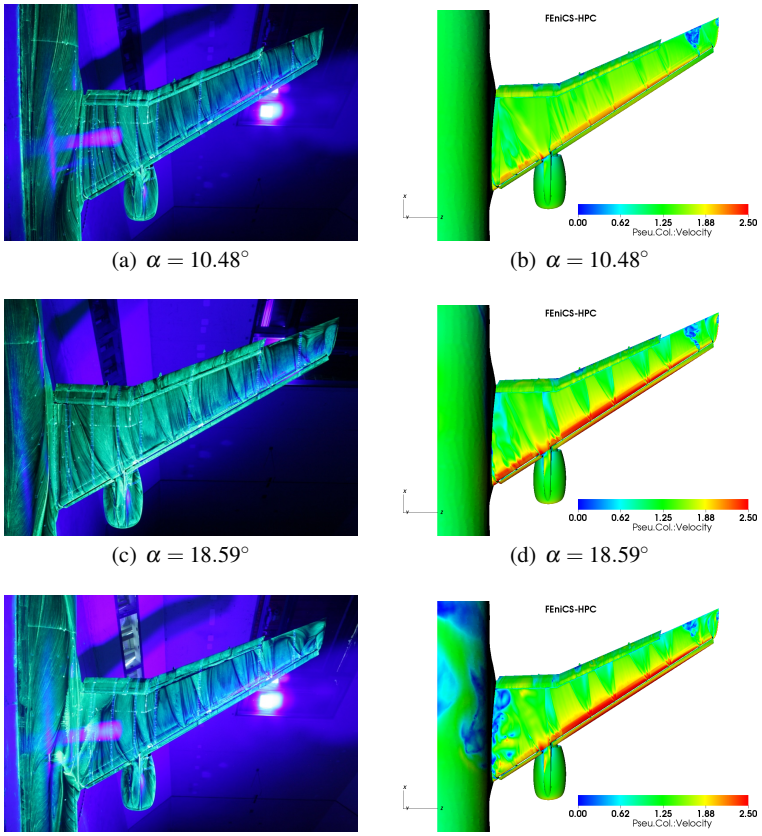


Fig. 10 Pressure coefficients, cp, versus normalized local chord, x/c, for the angle of attack $\alpha = 22.56^\circ$ untripped (left), the same angle $\alpha = 22.56^\circ$ tripped (middle) and $\alpha = 21.57^\circ$ tripped at locations A-A (top), D-D (middle) and G-G (bottom) for the wing of JSM pylon on.

tion of the dual Navier-Stokes equations. We begin by showing a plot of a volume rendering of the dual solution, see Figure 13.

What is worth noting here is that the adjoint velocity flows backwards in time and, consequently, it appears to be flowing in the opposite direction of the primal velocity. We observe that the part of the mesh where the dual velocity has higher values is *upstream* to the airplane. Because of the way the do-nothing error estimator is designed, we expect that the refinement will happen where both the residual and the dual solution are large. Indeed, this has the important implication that the mesh refinement will not only happen on the wing, where the forces are computed, but also upstream, splitting cells that, a priori, are unrelated to the computation of the aerodynamic forces.

This feature is unique for our methodology: while other methods tend to refine the mesh in zones where *intuitively* higher accuracy would yield better approximation of the aerodynamic forces, namely around the body and downstream, the adaptive algorithm provides an automatic procedure that knows nothing about the



(a) $\alpha = 10.48^\circ$ (b) $\alpha = 10.48^\circ$

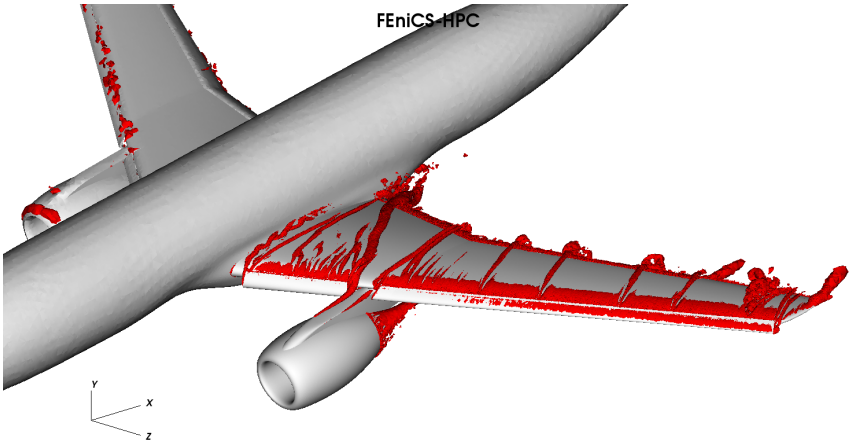
(c) $\alpha = 18.59^\circ$ (d) $\alpha = 18.59^\circ$

(e) $\alpha = 21.57^\circ$ (f) $\alpha = 22.56^\circ$

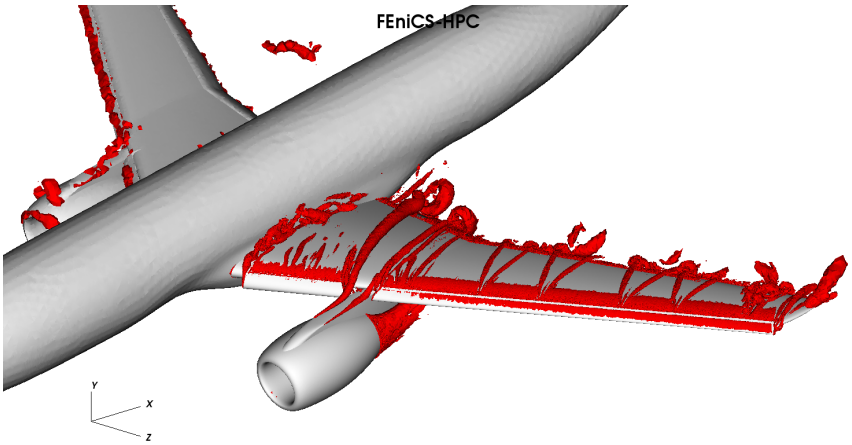
Fig. 11 Comparison between experimental oil film visualization (left) and surface rendering of the velocity magnitude (right).

features of the flow but only takes into account the residual of the equations of motion and the solution of the dual problem.

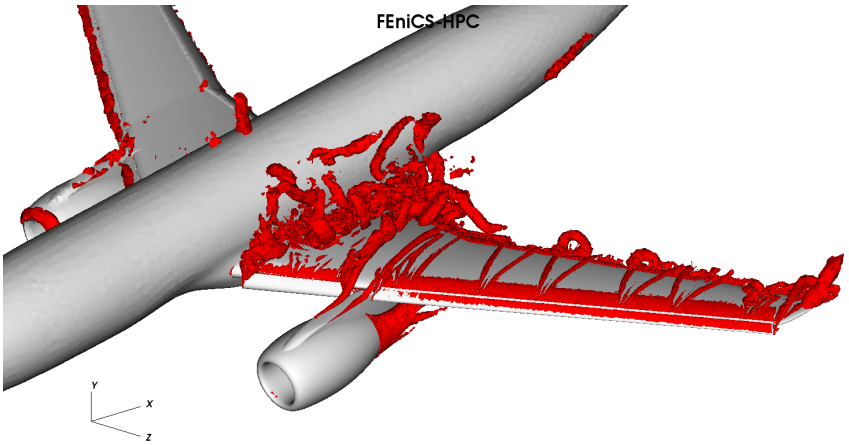
In our numerical experimentation we found that this is exactly what happens, as we are about to show. Consider Figure 14, showing a crinkled slice of the mesh for the initial and the finest meshes for a given angle of attack. It is clear that the mesh refinement procedure is concentrating both on the area around the surface where the aerodynamic forces are computed *and* in the upstream region. Some cells are refined downstream due to the large residual.



(a) $\alpha = 10.48°$



(b) $\alpha = 18.59°$



(c) $\alpha = 22.56°$

Fig. 12 Isosurface rendering of the Q-criterion with value $Q = 100$.

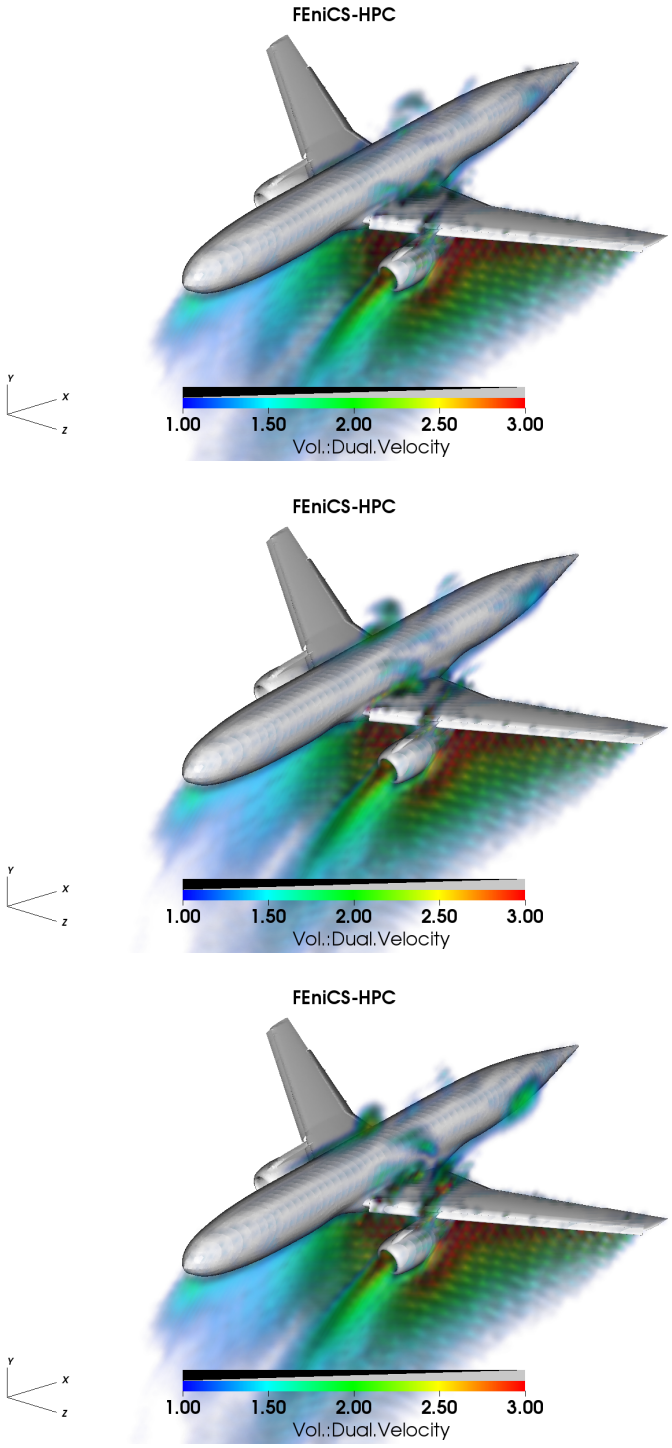
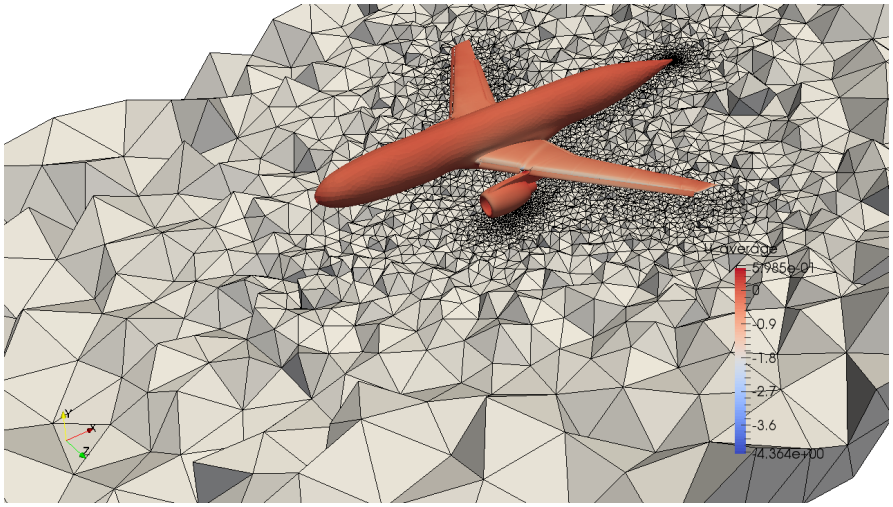
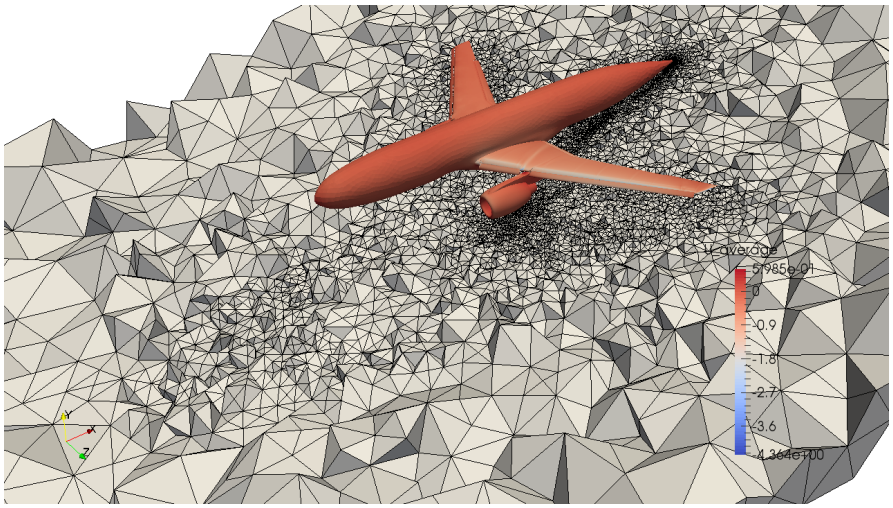


Fig. 13 Volume rendering of the time evolution of the magnitude of the adjoint velocity φ magnitude, snapshots at t = (16, 18, 20).



(a) Starting mesh



(b) Finest adaptive mesh

Fig. 14 Crinkled slice aligned with the angle of attack, $\alpha = 10.48^\circ$

4 Conclusions

This paper presents an adaptive finite element method without turbulence model parameters for time-dependent aerodynamics, and we validate the method by simulation results of a full aircraft model originating from the 3rd AIAA CFD High-Lift Prediction Workshop (HiLiftPW-3) which was held in Denver, Colorado, on June 3rd-4th 2017. The mesh is automatically constructed by the method as part of the

computation through duality-based a posterori error control and no explicit turbulence model is used. Dissipation of turbulent kinetic energy in under-resolved parts of the flow is provided by the numerical stabilization in the form of a weighted least squares method based on the residual of the NSE. Thus, the method is purely based on the NSE mathematical model, and no other modeling assumptions are made.

The DFS method and these simulations are thus *parameter-free*, where no a priori knowledge of the flow is needed during the problem formulation stage, nor during the mesh generation process. Additionally, the computational cost is drastically reduced by modeling turbulent boundary layers in the form of a slip boundary condition, and thus no boundary layer mesh is needed.

The computed aerodynamic coefficients, are very close to the experimental values for all the angles of attack that we studied. In particular, *cl* is within circa 5 % of the experiments, *cd* has a small overprediction of circa 10 %, which is consistent with the majority of the participants in HiLiftPW-3 across a range of methods Rumsey (2017), suggesting a systematic error in the problem statement or the experimental data.

The fact that the error is automatically estimated by the method is itself a critical feature missing in most (if not all) other computational frameworks for CFD.

Moreover, the adaptive procedure in DFS is seen to converge to a mean value with oscillations of the order of 1 % to 2 %. This contributes to increase the confidence in the numerical method.

The point of adaptive computations is all about saving on the computational cost. During the workshop we had the chance to compare our performance with that of the other participating groups. In terms of number of degrees of freedom, DFS is about ten times cheaper than the leading RANS and Lattice Boltzmann Methods.

To capture stall, we applied a tripping noise term that turned out to have the effect of triggering the physically correct stall separation pattern. A similar idea with a noise term is employed in the DNS community as well, and the addition of this term seems to have no effect on non-stalling configurations, which is an important validation.

We observed that DFS was able to capture the stall mechanism of the proposed configuration, namely the large scale separation pattern that occurs at the wing-body juncture. The same mechanism is observed in the experiments. The stall angle is also captured within ca. 1 °.

Acknowledgments.

This research has been supported by the European Research Council, the H2020 MSO4SC grant, the Swedish Research Council, the Swedish Foundation for Strategic Research, the Swedish Energy Agency, the Basque Excellence Research Center (BERC 2014-2017) program by the Basque Government, the Spanish Ministry of Economy and Competitiveness MINECO: BCAM Severo Ochoa accreditation SEV-2013-0323 and the Project of the Spanish Ministry of Economy and Competitiveness with reference MTM2013-40824 and La Caixa. We acknowledge the

Swedish National Infrastructure for Computing (SNIC) at PDC – Center for High-Performance Computing for awarding us access to the supercomputer resources Beskow.

References

FEniCS (2003), 'Fenics project', `http://www.fenicsproject.org`.

Hoffman, J. (2005), 'Computation of mean drag for bluff body problems using adaptive dns/les', *SIAM J. Sci. Comput.* **27(1)**, 184–207.

Hoffman, J. (2006), 'Adaptive simulation of the turbulent flow past a sphere', *J. Fluid Mech.* **568**, 77–88.

Hoffman, J. (2009), 'Efficient computation of mean drag for the subcritical flow past a circular cylinder using general galerkin g2', *Int. J. Numer. Meth. Fluids* **59(11)**, 1241–1258.

Hoffman, J., Jansson, J., de Abreu, R. V., Degirmenci, N. C., Jansson, N., Müller, K., Nazarov, M. & Spühler, J. H. (2013), 'Unicorn: Parallel adaptive finite element simulation of turbulent flow and fluid-structure interaction for deforming domains and complex geometry', *Comput. Fluids* **80**(0), 310 – 319.

Hoffman, J., Jansson, J., Degirmenci, C., Jansson, N. & Nazarov, M. (2012), *Unicorn: a Unified Continuum Mechanics Solver*, Springer, chapter 18.

Hoffman, J., Jansson, J. & Jansson, N. (2015), 'Fenics-hpc: Automated predictive high-performance finite element computing with applications in aerodynamics', *Proceedings of the 11th International Conference on Parallel Processing and Applied Mathematics, PPAM 2015. Lecture Notes in Computer Science* .

Hoffman, J., Jansson, J., Jansson, N., Johnson, C. & de Abreu, R. V. (2011), Turbulent flow and fluid-structure interaction, *in* 'Automated Solutions of Differential Equations by the Finite Element Method', Springer.
URL: *http://www.fenicsproject.org/pub/documents/book/*

Hoffman, J., Jansson, J., Jansson, N. & Nazarov, M. (2011), Unicorn: A unified continuum mechanics solver, *in* 'Automated Solutions of Differential Equations by the Finite Element Method', Springer.
URL: *http://www.fenicsproject.org/pub/documents/book/*

Hoffman, J., Jansson, J., Jansson, N., Vilela De Abreu, R. & Johnson, C. (2016), 'Computability and adaptivity in cfd. encyclopedia of computational mechanics, stein, e., de horz, r. and hughes, tjr eds'.

Hoffman, J., Jansson, J. & Stöckli, M. (2011), 'Unified continuum modeling of fluid-structure interaction', *Mathematical Models and Methods in Applied Sciences* .

Hoffman, J. & Jansson, N. (2010), *A computational study of turbulent flow separation for a circular cylinder using skin friction boundary conditions*, Ercoftac, series Vol.16, Springer.

Hoffman, J. & Johnson, C. (2006*a*), *Computational Turbulent Incompressible Flow: Applied Mathematics Body and Soul Vol 4*, Springer-Verlag Publishing.

Hoffman, J. & Johnson, C. (2006*b*), 'A new approach to computational turbulence modeling', *Comput. Methods Appl. Mech. Engrg.* **195**, 2865–2880.

Hoffman, J. & Johnson, C. (2007), *Computational Turbulent Incompressible Flow*, Vol. 4 of *Applied Mathematics: Body and Soul*, Springer.

Hoffman, J. & Johnson, C. (Published Online First at www.springerlink.com: 10 December 2008), 'Resolution of d'alembert's paradox', *J. Math. Fluid Mech.* .

Houzeaux, G., Vázquez, M., Aubry, R. & Cela, J. (2009), 'A massively parallel fractional step solver for incompressible flows', *Journal of Computational Physics* **228**(17), 6316–6332.

Huang, L., Huang, P. G. & LeBeau, R. P. (2004), 'Numerical study of blowing and suction control mechanism on naca 0012 airfoil', *AIAA Journal of aircraft* **41**(1).

Hunt, J. C., Wray, A. A. & Moin, P. (1988), 'Eddies, streams, and convergence zones in turbulent flows'.

Jansson, N., Hoffman, J. & Jansson, J. (2012), 'Framework for Massively Parallel Adaptive Finite Element Computational Fluid Dynamics on Tetrahedral Meshes', *SIAM J. Sci. Comput.* **34**(1), C24–C41.

Kirby, R. C. (2012), *FIAT: Numerical Construction of Finite Element Basis Functions,*, Springer, chapter 13.

Logg, A., Mardal, K.-A., Wells, G. N. et al. (2012), *Automated Solution of Differential Equations by the Finite Element Method*, Springer.

Logg, A., Ølgaard, K. B., Rognes, M. E. & Wells, G. N. (2012), *FFC: the FEniCS Form Compiler*, Springer, chapter 11.

Mellen, C. P., Frölich, J. & Rodi, W. (2003), 'Lessons from lesfoil project on large-eddy simulation of flow around an airfoil', *AIAA journal* **41**, 573–581.

Moin, P. & You, D. (2008), 'Active control of flow separation over an airfoil using synthetic jets', *Journal of Fluids and Structures* **24**(8), 1349–1357.

Piomelli, U. & Balaras, E. (2002), 'Wall-layer models for large-eddy simulation', *Annu. Rev. Fluid Mech.* **34**, 349–374.

Rumsey, C. (2017), '3^{rd} AIAA CFD High Lift Prediction Workshop (HiLiftPW-2) (http://hiliftpw.larc.nasa.gov/)'.
URL: *http://hiliftpw.larc.nasa.gov/*

Sagaut, P. (2005), *Large Eddy Simulation for Incompressible Flows (3rd Ed.)*, Springer-Verlag, Berlin, Heidelberg, New York.

Schlatter, P. & Orlu, R. (2012), 'Turbulent boundary layers at moderate reynolds numbers: inflow length and tripping effects', *Journal of Fluid Mechanics* **710**, 534.

Shan, H., Jiang, L. & Liu, C. (2005), 'Direct numerical simulation of flow separation around a naca 0012 airfoil', *Computers and Fluids* **34**, 10961114.

Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E. & Mavriplis, D. (2014), 'Cfd vision 2030 study: a path to revolutionary computational aerosciences'.

Spalart, P. R. (2009), 'Detached-eddy simulation', *Annu Rev. Fluid Mech.* **41**, 181–202.

Vilela de Abreu, R., Jansson, N. & Hoffman, J. (2014), 'Adaptive computation of aeroacoustic sources for a rudimentary landing gear', *Int. J. Numer. Meth. Fluids*

74(6), 406–421.
URL: *http://dx.doi.org/10.1002/fld.3856*

Witherden, F. D. & Jameson, A. (2017), Future directions of computational fluid dynamics, *in* '23rd AIAA Computational Fluid Dynamics Conference', p. 3791.

Received: 5 November 2018 | Revised: 27 May 2019 | Accepted: 20 June 2019

DOI: 10.1002/cnm.3232

RESEARCH ARTICLE

WILEY

A computational model of open-irrigated radiofrequency catheter ablation accounting for mechanical properties of the cardiac tissue

Argyrios Petras[1] | **Massimiliano Leoni**[1,2] | **Jose M. Guerra**[3] | **Johan Jansson**[1,2] | **Luca Gerardo-Giorda**[1]

[1]Basque Center for Applied Mathematics, Bilbao, Spain

[2]Department of Computational Science and Technology, KTH Royal Institute of Technology, Stockholm, Sweden

[3]Department of Cardiology, Hospital de la Santa Creu i Sant Pau, Barcelona, Spain

Correspondence
Luca Gerardo-Giorda, Basque Center for Applied Mathematics, Alameda de Mazarredo 14, 48009 Bilbao, Spain.
Email: lgerardo@bcamath.org

Funding information
"la Caixa" Foundation, Grant/Award Number: 2016 PhD grant to M. Leoni; Abbott Laboratories, Grant/Award Number: non conditional grant to J.M. Guerra; Eusko Jaurlaritza, Grant/Award Number: BERC 2013-2017 and BERC 2018-2021; H2020 Energy, Grant/Award Number: MSO4SC; Spanish State Research Agency (AEI), Grant/Award Number: SEV-2017-0718, RTI2018-093416-B-I00; Ministerio de Economía y Competitividad, Grant/Award Number: MTM2015-69992-R and MTM2016-76016-R

Abstract

Radiofrequency catheter ablation (RFCA) is an effective treatment for cardiac arrhythmias. Although generally safe, it is not completely exempt from the risk of complications. The great flexibility of computational models can be a major asset in optimizing interventional strategies if they can produce sufficiently precise estimations of the generated lesion for a given ablation protocol. This requires an accurate description of the catheter tip and the cardiac tissue. In particular, the deformation of the tissue under the catheter pressure during the ablation is an important aspect that is overlooked in the existing literature, which resorts to a sharp insertion of the catheter into an undeformed geometry. As the lesion size depends on the power dissipated in the tissue and the latter depends on the percentage of the electrode surface in contact with the tissue itself, the sharp insertion geometry has the tendency to overestimate the lesion obtained, which is a consequence of the tissue temperature rise overestimation. In this paper, we introduce a full 3D computational model that takes into account the tissue elasticity and is able to capture tissue deformation and realistic power dissipation in the tissue. Numerical results in FEniCS-HPC are provided to validate the model against experimental data and to compare the lesions obtained with the new model and with the classical ones featuring a sharp electrode insertion in the tissue.

KEYWORDS

elastic tissue deformation, finite elements, open-irrigated catheter, radiofrequency ablation

1 | INTRODUCTION

Radiofrequency catheter ablation (RFCA) is a minimally invasive therapy widely used for the treatment of various types of cardiac arrhythmias.[1,2] Typically, for endocardial radiofrequency ablations (RFAs), a catheter is advanced into the cardiac chamber from the groin of the patient through a blood vessel. Radiofrequency (RF) current at 500 kHz is delivered to the arrhythmogenic tissue via the electrodes on the catheter, producing resistive electrical heating in the neighborhood of the electrode. Conduction then propagates the heating through the rest of the tissue. At a target temperature, usually of 50°C, irreversible damage is inflicted to the tissue, and a lesion is formed.[1]

The RFA is considered an effective and safe treatment for cardiac arrhythmias; however, a number of life threatening complications may occur. The rise of blood temperature between 50°C and 80°C may lead to the formation of soft thrombi around the electrode and the heated tissue and is suspected to be associated with thromboembolic events.[3,4] In addition, excessive heating of the blood near the electrode at 80°C and above leads to the denaturation of the blood proteins and the formation of a coagulum at the tip of the catheter.[1,3] Finally, extreme heating of the tissue at 100°C leads to steam formation and audible steam pops occur,[5-7] which could result in myocardial tear or tamponade.[1]

Several computational models have been developed to describe the biophysics of the cardiac RFA treatment process. Some use an axisymmetric approach and develop 2D geometries, on which they employ Pennes' bioheat equation with convective boundary conditions for the cooling effect of the blood flow, coupled with a quasi-static electrical potential equation.[8-11] To model the effect of the coolant liquid of open-irrigated catheters, these models typically fix the temperature of the tip electrode.[12,13]

Three-dimensional cardiac RFA models that use the same mathematical equations as the 2D axisymmetric models are also available.[14-16] For open-irrigated catheters, some models include the interaction of the saline inflow with a layer of saline in epicardial modelling[17] or with the surrounding blood in endocardial RFA.[18] Typically, a sharp insertion of the catheter in the tissue is considered, disregarding its elastic deformation. In particular, the catheter is unphysically penetrating the tissue without deforming it, as shown in the bottom row of Figure 6. However, the need for the incorporation of mechanical deformation in RFA modelling was hinted in a previous work.[16] Indeed, the amount of power delivered to the tissue depends on the surface of contact between the electrode and the tissue itself.[19] Sharp insertion produces larger surfaces of contact and is likely to overestimate the actual effect of the RFA treatment. The first RFA model that includes elastic deformation of the tissue uses profiles that are extracted from X-ray scanning before the ablation process.[20] No other work was found in our literature review that includes the tissue deformation because of the contact with the catheter.

We propose a computational model that includes both physical and mechanical properties of the tissue. We consider in this work a six-hole open electrode with a hemispherical tip, placed vertically on the cardiac tissue. We model the elastic deformation of the tissue in contact with the tip by solving the axisymmetric Boussinesq problem for a spherical punch. We use a modified version of Pennes' bioheat equation to model the thermal effect of RFA and a quasi-static electrical potential equation. Constant-power ablation is obtained by augmenting the system with a power constraint equation. The model also features the incompressible Navier-Stokes equations to describe the flow of blood and the saline flow from the open-irrigated electrode. We validate our proposed computational model against a set of suitably defined in vitro experiments performed at J. M. Guerra's lab at the Hospital de la Santa Creu i San Pau in Barcelona. We investigate the effect of tissue elasticity in the computational RFA models by comparing the computational lesion dimensions of the deformed tissue case against the undeformed one that is typically used in state-of-the-art models.

The paper unfolds as follows. Section 2 describes our mathematical model, and in Section 3, we describe its calibration. In Section 4, we validate the proposed model against a set of in vitro experiments: we describe the experimental setup and discuss the results of the proposed computational model. We also compare the results of our model with the ones obtained with a geometry featuring a sharp insertion of the electrode. Finally, in Section 5, we discuss our findings and the limitations of the proposed model and suggest some ways to overcome them.

2 | COMPUTATIONAL MODEL

Our mathematical model consists of a system of partial differential equations (PDEs) that describe the evolution of the temperature, the flow of blood, and the cooling saline solution, as well as the electrical field generated by the RFA procedure. In this section, we expand on its main characteristics: the geometry, the governing equations, and the method to assess the computational lesion.

2.1 | Geometry

The computational domain that we consider in this paper is based on an in vitro experimental setup, which is similar to the one presented by Guerra et al.[2] We construct a box of dimensions $80 \times 80 \times 80$ mm that consists of the blood chamber $80 \times 80 \times 40$ mm, the cardiac tissue $80 \times 80 \times 20$ mm, and a board $80 \times 80 \times 20$ mm underneath the tissue. The board separates the tissue from the dispersive electrode, which is located at the bottom of the computational domain. This is an important aspect, matching the settings in the experiments used for validation, where the tissue is not in direct contact with the dispersive electrode but is actually placed on a physical board of methacrylate. Moreover, in clinical RFA, the dispersive electrode is placed on a patch on the back (or thigh) of the patient. Having the dispersive electrode in direct

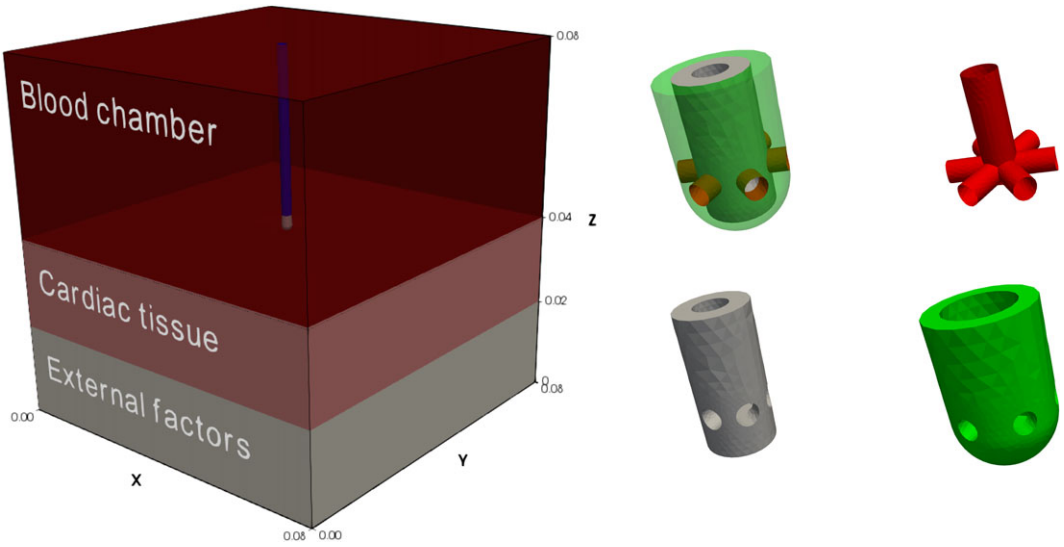


FIGURE 1 The full computational geometry (left) and detailed view of the catheter tip and its components (right): The assembled catheter tip, the saline pipes, the thermistor, and the electrode (Top left to bottom right)

contact with the tissue would lead to an excessive simplification of the model. Also, the presence of the board allows us to match the impedance of the RFA system (provided by the machine) without modifying the electrical conductivity of the tissue. This electrical conductivity was tuned to match the initial impedance of the system in a previous work,[18] thus altering the biophysical properties of the tissue with respect to what is reported in the specialized literature. Our approach ensures the correct biophysical characterization of the tissue, summarized by the parameters in Section 3.1 for porcine myocardium.

We place a catheter perpendicular to the tissue at the center of the box inside the blood chamber, causing a deformation that depends on the catheter-tissue contact force (see Section 2.5). Figure 1 (left) shows the described computational geometry. We consider a six-hole open-irrigated electrode of diameter of 2.33 mm and length of 3.5 mm with a hemispherical tip. The electrode contains a thermistor of diameter of 1.54 mm and length of 3 mm. The holes, of diameter of 0.5 mm, are connected to an inner channel of diameter of 0.73 mm inside the thermistor, which is connected to the catheter body and allows the saline to flow in the blood chamber. Figure 1 (right) shows the geometry of the open-irrigated electrode.

The computational domain consists of five different subdomains: the blood chamber (Ω_{blood}), the cardiac tissue (Ω_{tissue}), the external factors board (Ω_{board}), the electrode (Ω_{el}), and the thermistor (Ω_{therm}). In what follows, we will simply denote

$$\Omega := \Omega_{blood} \cup \Omega_{tissue} \cup \Omega_{board} \cup \Omega_{el} \cup \Omega_{therm}.$$

The catheter body and the inner tubes at the open-irrigated electrode are considered thermally and electrically insulated, and therefore, insulation boundary conditions are applied. Additional details on the thermal and electrical boundary conditions are provided in Sections 2.3 and 2.4.

2.2 | Blood and saline flow

The open-irrigated electrode allows the cooling saline to flow out through the holes at its tip and mix with the blood. The comparable densities between saline and blood justifies the assumption of a perfect mix in the blood chamber. The flow dynamics is thus governed by the incompressible Navier-Stokes equations:

$$\begin{aligned} \frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \operatorname{div} \sigma(\mathbf{u}, p) &= \mathbf{0} \\ \operatorname{div} \mathbf{u} &= 0, \end{aligned} \tag{1}$$

where $\mathbf{u}$ is the flow velocity, p is the pressure scaled by the density ρ, and $\sigma(\mathbf{u}, p) = 2\mu\rho^{-1}\frac{\nabla\mathbf{u}+\nabla\mathbf{u}^T}{2} - pI$ is the stress tensor, μ being the dynamic viscosity of the blood.

We impose zero pressure and stress as an outflow boundary condition on the right side of the domain $\{x = 0.08\}$. A constant inflow boundary condition is imposed on the left side of the domain $\{x = 0\}$,

$$\mathbf{u} = \mathbf{u}_{in} = (u_b, 0, 0),$$

and on the electrode holes for the flow of the coolant liquid,

$$\mathbf{u} = u_s \mathbf{n}_e,$$

where u_b and u_s are the magnitudes of the velocity of the blood and the coolant liquid respectively, while $\mathbf{n}_e$ is the unit normal vector of the electrode pointing toward the blood chamber. All the remaining boundaries, including the internal one separating the blood chamber from the tissue, are equipped with no-slip boundary conditions. See Figure 3 for a representation of all boundary conditions.

2.3 | Electrical field

The electrical field is generated by the electrode at the tip of the catheter. At a frequency of 500 kHz, and over the distance of interest, the biological medium can be considered totally resistive, and the electrical problem can be set in quasi-static form:

$$\operatorname{div}(\sigma(T)\nabla\Phi) = 0, \tag{2}$$

where Φ is the electrical potential. We consider the electrical conductivity $\sigma = \sigma(T)$ (not to be confused with the stress tensor $\sigma(\mathbf{u}, p)$ introduced in Equation (1)) to be constant in all regions but in the tissue, where we consider a linear dependency on the temperature (see Section 3.1).

For a constant-power ablation mode, Equation (2) is augmented with a constant-power constraint[21]

$$\int_\Omega \sigma(T)\nabla\Phi \cdot \nabla\Phi \, \mathrm{d}x = P, \tag{3}$$

where Ω is the computational domain and P is the constant power imposed.

We set an initial potential of 0 V everywhere. We impose a zero potential condition on the bottom of the domain to model the dispersive electrode, while on the remaining external surfaces and the catheter boundary, we assign an insulation condition ($-\sigma\nabla\Phi \cdot n = 0$).

2.4 | Tissue heating

The application of an electrical potential at the tip electrode of the catheter produces resistive heating at the cardiac tissue and the surrounding blood. A modification of Pennes' bioheat equation models both the heating by the direct application of RF current and the conductive heating[18] and reads the following:

$$\frac{\partial(\rho h)}{\partial t} - \operatorname{div}(k(T)\nabla T) = q - \rho c(T)\mathbf{u} \cdot \nabla T + Q_m - Q_p, \tag{4}$$

where ρ is the density, h is the specific enthalpy, $c(T)$ is the specific heat, $k(T)$ is the thermal conductivity of the medium, $q = \sigma(T)|\nabla\Phi|^2$ is the distributed heat source from the electrical field, Q_m is the metabolic heat generation, and Q_p is the heat loss because of the blood perfusion. For endocardiac RFA, the last two terms in this equation can be omitted for short ablation times.[15,18,22]

In biological tissues, the specific enthalpy h and the temperature T are related as follows:

$$\frac{\partial(\rho h)}{\partial t} = \psi(T)\frac{\partial T}{\partial t},$$

where $\psi(T) = \rho c(T)$. Finally, we assume the thermal conductivity $k(T)$ and the specific heat $c(T)$ to be piecewise constant, with different values in different regions of the domain, except in the tissue, where we consider a linear dependence on the temperature (see Section 3.1).

The final form of the bioheat equation is then as follows:

$$\rho c(T)\left(\frac{\partial T}{\partial t}+\mathbf{u}\cdot\nabla T\right)-\operatorname{div}(k(T)\nabla T)=\sigma(T)|\nabla\Phi|^2. \tag{5}$$

2.5 | Elasticity

During the ablation procedure, the electrode at the tip of the catheter applies a contact force to the cardiac tissue, resulting in its deformation. For a catheter with a hemispherical tip such as the one we consider in this paper (see Figure 1), the problem is equivalent to a spherical indenter on an elastic half-space in contact mechanics. There are many different approaches for this contact mechanics problem, including Hertzian theory for shallow indentations,[23,24] a modified Hertzian solution,[25] and the axisymmetric Boussinesq problem.[26] We use the axisymmetric Boussinesq problem approach as it provides a general framework for different profiles, which will allow us to consider also different catheter tips in the future. In this work, we consider a spherical indenter profile.

Following Sneddon,[26] because of the radial symmetry of the contact problem, consider the two-dimensional problem on the vertical z-axis and the horizontal r-axis, which represents the distance from the center of the sphere. The deformed elastic tissue follows the spherical indenter from the center of the sphere, where the maximum vertical displacement $\omega_{\max}$ occurs, up to the contact radius a. After the contact point, the elastic tissue deformation is smoothly decreasing as the distance r increases. Figure 2 shows the deformation of the tissue, as described.

For the case $r \leq a$, the elastic tissue follows the sphere, ie,

$$r^2+[z+(R-\omega_{\max})]^2=R^2,$$

where R is the radius of the sphere. The positive solution of this system for z gives the displacement of the interface $\omega(r)$ as a function of the distance r,

$$\omega(r)=\omega_{\max}-f(r/a), \tag{6}$$

where $f(t)=R-\sqrt{R^2-a^2t^2}$. The function f is the profile of the spherical indenter that is used in the rest of the computations.

The maximum depth is given by:

$$\omega_{\max}=\frac{a}{2}\log\left(\frac{R+a}{R-a}\right), \tag{7}$$

while the total force is:

$$F=\frac{G}{1-\nu}\left((a^2+R^2)\log\left(\frac{R+a}{R-a}\right)-2aR\right), \tag{8}$$

where ν is the Poisson's ratio and

$$G=\frac{E}{2(1+\nu)}$$

is the shear modulus of the tissue, E being the Young's modulus of elasticity.

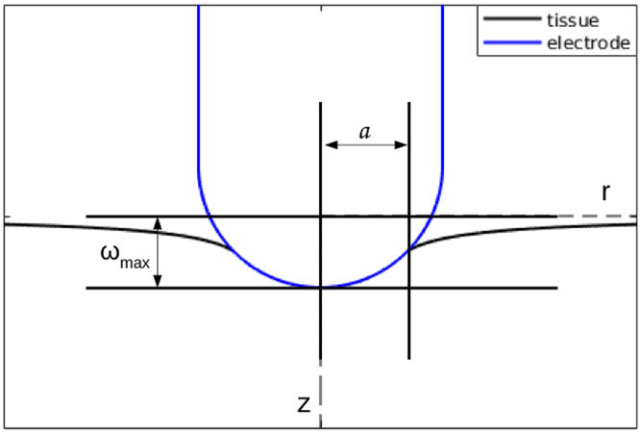


FIGURE 2 A cross section of the electrode-tissue contact

For $r > a$, the deformation can be calculated by evaluating the integral:

$$\omega(r) = \int_0^1 \frac{\chi(t)}{(r/a)^2 - t^2}\, dt, \tag{9}$$

where

$$\chi(t) = \frac{2\omega_{\max}}{\pi} - \frac{at}{\pi} \log\left(\frac{R+at}{R-at}\right).$$

Thus, given the contact force F, the contact radius a can be calculated by Equation (8) using optimization techniques. The remaining quantities can be calculated using the corresponding formulas above. The displacement of the interface $\omega(r)$ is calculated using numerical integration techniques for $r > a$.

2.6 | Model summary

We consider a 3D computational geometry, as described in Section 2.1, and we compute the vertical displacement of the surface of the tissue for a given catheter contact force (see Section 2.5).

We summarize the mathematical model in the following system of coupled PDEs:

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} - \operatorname{div} \sigma(\mathbf{u}, p) = \mathbf{0} \quad \text{in} \quad \Omega_{blood} \times (0, T), \tag{10a}$$

$$\operatorname{div} \mathbf{u} = 0 \quad \text{in} \quad \Omega_{blood} \times (0, T), \tag{10b}$$

$$\rho c(T) \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T\right) - \operatorname{div}(k(T) \nabla T) = \sigma(T) |\nabla \Phi|^2 \quad \text{in} \quad \Omega \times (0, T), \tag{10c}$$

$$\operatorname{div}(\sigma(T) \nabla \Phi) = 0 \quad \text{in} \quad \Omega \times (0, T). \tag{10d}$$

The above equations are equipped with a number of different boundary conditions, applied to the various parts of our computational geometry. Figure 3 collects the boundary conditions of (10a) to (10d): unless stated otherwise, we impose zero velocity ($\mathbf{u} = \mathbf{0}$), zero current flux ($\sigma \nabla \Phi \cdot n = 0$), and body temperature ($T = T_b = 37°\text{C}$) on all the surfaces of the boundary. We impose zero velocity on the tissue surface, which is part of the boundary of Ω_{blood}. The values of the saline inlet $\mathbf{u}_s$ are given by the RF protocol under consideration. The homogeneous Dirichlet boundary condition for the potential $\Phi = 0$ on the bottom surface models the dispersive electrode, while the Dirichlet condition $\Phi = V_0$ on the upper boundary of the electrode is discussed in the upcoming Sections 3.1.2 and 3.1.3. We set the initial conditions to $p = 0$, $\mathbf{u} = \mathbf{0}$, $\Phi = 0$, and $T = T_b$ in the whole domain.

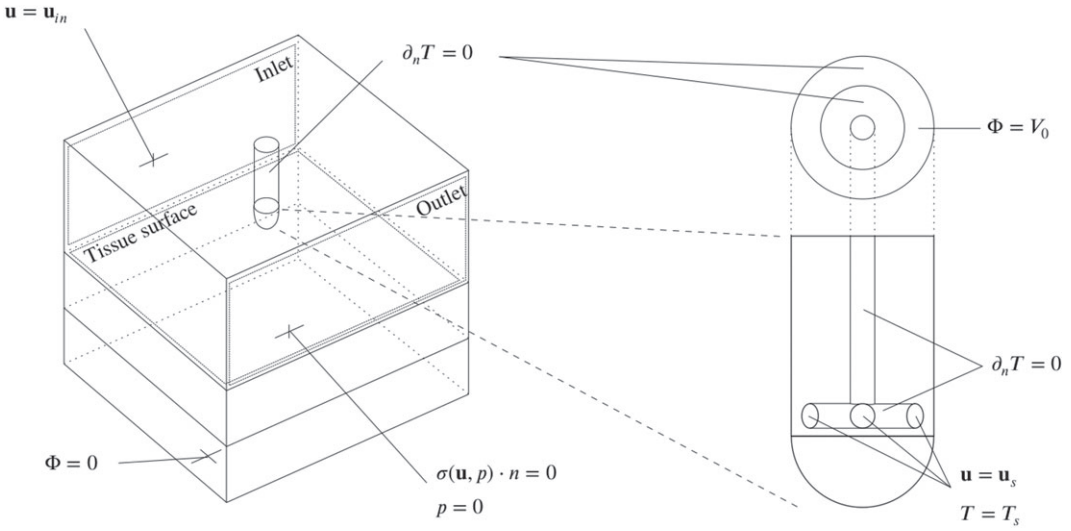


FIGURE 3 Boundary conditions that differ from zero velocity ($\mathbf{u} = \mathbf{0}$), zero current flux ($\sigma \nabla \Phi \cdot n = 0$), and body temperature ($T = T_b$)

2.7 | Lesion assessment

The prediction of the lesion size is important to ensure the effectiveness of RF treatment. During the ablation process, RF current flows through the myocardium and via Joule heating, a rise in the tissue temperature is produced in a small area around the electrode tip. The tissue is irreversibly damaged, and a thermal lesion is formed when the temperature rises above 50°C.[1] In light of the above consideration, we assume that a lesion is fully formed at the isotherm contour of 50°C. In order to assess the lesion characteristics, typical quantities of interest are its depth (D), width (W), depth at which the maximal width occurs (DW), and its volume (V). Also, the area of the lesion on the tissue surface (S) is important to study the possible formation of soft thrombi.[3] The characteristic dimensions we consider in this paper to validate the computational model are described in Section 4.2 and collected in Figure 4 therein.

3 | CALIBRATION OF THE MODEL

3.1 | Model parameters

The parameters considered in this paper are drawn from the literature and are summarized in Table 1. Specifically, the physical properties of the blood are obtained from the virtual population database from the IT IS foundation.[27] The electrode and thermistor physical properties were found in the literature.[16,18] We assume the electrical conductivity of the external factors board σ_b to be constant, and we tune it to match the initial impedance of the experimental setup, as detailed in Section 3.1.2. We choose the remaining parameters for the board to be the same as the initial tissue state; they appear to have no impact on the results of our numerical simulations.

Considering that in our in vitro experiments, RFA was performed on a porcine myocardium, we used the corresponding physical properties.[28-30] We use a linear temperature dependence of specific heat, thermal, and electrical conductivities of the tissue according to the aforementioned literature:

$$c(T) = c_0(1 - 0.0042(T - 37)), \quad k(T) = k_0(1 - 0.0005(T - 37)), \quad \sigma(T) = \sigma_0(1 + 0.015(T - 37)),$$

where c_0, k_0, and σ_0 are the specific heat, thermal, and electrical conductivities at body temperature (see Table 1). Finally, since the experiments were performed in an in vitro condition, it is reasonable to assume that the cardiac tissue got stiffer during preparation. We thus used in our simulations the Young's modulus of the cardiac tissue at systole,[31] which is within the range of measurements in an in vitro experimental setup found in Ramadan.[32]

3.1.1 | Tissue power delivery calibration

In our model, we set the electrical conductivity of the board σ_b to match the power dissipated in the tissue P_{tissue}, that we identify following the analysis of Wittkampf and Nakagawa in Wittkampf and Nakagawa[19]: the percentage of the total power P delivered to the tissue is given by the following formula:

$$P_{tissue} = \frac{A_{tissue}\sigma_{tissue}}{A_{blood}\sigma_{blood} + A_{tissue}\sigma_{tissue}} P =: \alpha P, \tag{11}$$

where P is the total power set by the machine in the experiment and A_{blood} and A_{tissue} is the contact area of the electrode with the blood and the tissue, respectively.

TABLE 1 The parameters we considered in our proposed computational model

Parameter	Blood	Tissue	Electrode	Thermistor	Board
ρ (kg/m^3)	1050	1076	21 500	32	1076
c (J/kg/K)	3617	$c_0 = 3017$	132	835	3017
k (W/m/K)	0.52	$k_0 = 0.518$	71	0.038	0.518
σ (S/m)	0.748	$\sigma_0 = 0.54$	4.6×10^6	10^{-5}	σ_b
$\mu\rho^{-1}$ (m^2/s)	2.52×10^{-6}	–	–	–	–
ν (-)	–	0.499	–	–	–
E (kg/m/s^2)	–	75×10^3	–	–	–

In our computational geometry (see Section 2.1), we chose a hemispherical catheter tip. Thus, the surface area of the catheter in contact with the tissue A_{tissue} can be calculated as follows:

$$A_{tissue} = \begin{cases} 2\pi Rh, & \text{if } h \leq R, \\ 2\pi R^2 + 2\pi R(h-R) - 6R_h^2 \cos^{-1}\left(1 + \frac{R-h}{R_h}\right), & \text{if } R < h \leq R + R_h \\ 2\pi R^2 + 2\pi R(h-R) - 6R_h^2\left(\frac{\pi}{2} + \cos^{-1}\left(2 + \frac{R-h}{R_h}\right)\right), & \text{if } R + R_h < h \leq R + 2R_h \\ 2\pi R^2 + 2\pi R(h-R) - 6\pi R_h^2, & \text{if } h \geq R + 2R_h, \end{cases},$$

where $h = \omega_{\max} - \omega(a)$ is the contact depth of the catheter with the tissue (see Section 2.5) and R_h is the radius of the irrigation holes on the catheter (see Section 2.1). The blood surface area is calculated using the following formula:

$$A_{blood} = 2\pi R^2 + 2\pi R(h_e - R) - 6\pi R_h^2 - A_{tissue},$$

where h_e is the length of the electrode.

3.1.2 | External factors calibration

We included the external factors board in the model to represent the part of the experimental setup, between the tissue and the dispersive electrode, that is not modeled. The constitutive parameters of the board are the same as the tissue (see Table 1), with the exception of the electrical conductivity σ_b, which is tuned to match the actual power delivered to the tissue. This approach allows us to use as much of the information available from the RFA system as possible and provides us with great flexibility should different types of tissue be considered, in a major improvement with respect to standard approaches on the topic (including earlier works by some of the authors[18]) that tuned the tissue conductivity to match the overall system's initial impedance.

Given the total power of the ablation P and the initial impedance of the system R (both values available from the RFA system), the initial potential drop in the system is given by Ohm's law as $V_0 = \sqrt{PR}$. We can assume that the electrode cable is not dissipating energy; thus, we can consider the potential drop to occur entirely between the cable-electrode junction and the dispersive electrode. We thus impose a Dirichlet boundary condition on the upper boundary of the electrode (see Figure 3), and we solve the potential equation in the whole domain Ω. The advantage of such a framework is that the interfaces between different materials are actually internal discontinuity surfaces of the domain: the equation itself takes care of the conductances drop across internal layers as a consequence of the solution's continuities.[33] In addition, modifications of the electrode material would be immediate.

The calibration of the external factors board conductivity is an iterative preprocessing step. Given an initial guess of σ_b in the subdomain Ω_{board}, we solve Equation (10d) in the whole computational domain Ω, where σ is a shorthand for the electrical conductivity in each subdomain (Table 1). We impose $V_0 = \sqrt{PR}$ as boundary condition (see Figure 3) and iteratively update the value of σ_b, until the constraint,

$$\int_{\Omega_{tissue}} \sigma_0 |\nabla \Phi|^2 \, \mathrm{d}x = P_{tissue}, \tag{12}$$

is satisfied. In Equation (12), P_{tissue} is the one in Equation (11), and σ_0 is the electrical conductivity of the tissue at body temperature (see Table 1). In the code developed to obtain the results of Section 4, we implement a root-finding algorithm via bisection search, that stops once the difference between two subsequent iterations is smaller than 0.01 W. The calculated σ_b is kept constant throughout the simulation.

3.1.3 | Voltage calibration during the simulation

In Section 2.3, no boundary conditions were specified for the potential equation at the upper part of the electrode, but we required the solution to satisfy a constraint on the total dissipated power (3). However, because of the difference in size between our computational domain and the actual ablation site, the total power dissipated in the computational setting is expected to be smaller than the one provided by the machine. We assume the power dissipated outside our computational

domain as well as the one in Ω to be constant in time. During the simulation, we update the potential V_0 on the upper interface of the electrode so that the solution satisfies:

$$\int_{\Omega} \sigma |\nabla \Phi|^2 \, \mathrm{d}x = P_0. \tag{13}$$

In the above formula, P_0 is the total power dissipated in our computational domain at the beginning of the simulation, and σ is the corresponding electrical conductivity for each subdomain of the full computational domain Ω. In the subdomain Ω_{board}, the σ_b described in the previous section is used. We assume a constant voltage on the surface where we impose the boundary condition: the electrode is made of conductive material, and we know from standard theory of electromagnetism that electrons can move freely on the surface of a conductor, tending to a configuration of equilibrium.

4 | NUMERICAL RESULTS

For the simulations in this work, we built a 3D computational geometry for each given catheter contact force, where the vertical displacement of the surface of the tissue is computed by means of a Clenshaw-Curtis numerical integration method.[34]

The resulting domain Ω is discretized by a tetrahedral unstructured mesh, with adaptive refinement in the neighborhood of the electrode. The mesh is built with Salome[35] and is properly formatted using.[36,37] For all the simulations presented in this work, the meshes feature approximately 5 000 000 elements with minimum and maximum element sizes of the order of magnitude of 5×10^{-5} m and 1×10^{-3} m. Following the instructions of the catheter producer, the saline inflow is set at u_s =0.24 m/s, corresponding to an irrigation rate of 17 mL/min subdivided among the six irrigation holes. In the model validation section, the blood inflow velocity is set to $u_b = 0.5$ m/s to match our experimental setup described in the upcoming Section 4.2. In the comparison between elastic deformation and sharp insertion, two blood flow protocols are considered, as in the experiments by J. Guerra and his collaborators[2]: a high-flow (HF) protocol with $u_b = 0.5$ m/s, and a low-flow (LF) protocol, with $u_b = 0.1$ m/s.

4.1 | Simulations protocol

Computational modeling of ablation as well as other medical procedures and diagnostics in cardiovascular science present significant scientific challenges, such as robust turbulent fluid-structure interaction (FSI) with contact and heating, control of the computational error in multiphysics problems, and complex software implementation. For computational technology to be effective in overcoming these challenges, it has to be **efficient**: enabling solution of advanced problems with available resources; **reliable**: providing quantitative error control or the computational solution in a chosen output, as well as a guarantee of correctness of software implementation; and **accessible**: enabling scientists to access, use, and extend the simulation technology as a tool in their daily work.

We address these problems in the setting of a general stabilized adaptive finite element methodology we denote Direct FEM Simulation (DFS)[38] realized in the automated massively parallel FEniCS-HPC open source software framework,[39,40] taking the weak form of the PDE as input and automatically generating low-level source code for assembling tensors and duality based a posteriori error estimates and indicators for adaptive error control with good scaling on supercomputers. In this framework, we developed an RFA module in FEniCS-HPC explicitly for the present work to discretize and solve system (10a) to (10d).

Navier-Stokes equations

The Navier-Stokes equations are solved via the DFS method presented in Hoffman et al and Jansson et al,[38,41,42] consisting of a standard P1-P1 finite elements method with Galerkin least-squares stabilization and optional goal-oriented adaptive mesh refinement. The Navier-Stokes system is discretized in time by a Crank-Nicholson scheme, and DFS solves the full nonlinear problem via successive minimizations based on a fixed-point approach.[38,42]

Bioheat equation

The Bioheat equation, on the other hand, is solved with P1 elements and is advanced in time by a backward Euler scheme, where the coefficients are linearized around the previous time step. Since for the set of parameters introduced

in Section 3.1, the Bioheat equation is advection dominated in Ω_{blood}, a streamline-upwind Petrov-Galerkin (SUPG) stabilization term is added to the discrete equation.[43]

Potential equation

The potential equation is solved with P1 elements. The equation depends implicitly on time through the conductivity. As the simulation proceeds, the rise in the tissue temperature modifies its electrical conductivity, and the constraint on the total dissipated power will eventually fail to be satisfied. We thus enforce constraint (13) at each time step when we solve the potential equation as follows.

Let Φ_1 be the solution of

$$\begin{aligned} \operatorname{div}(\sigma^{new}\nabla\Phi_1) &= 0 && \text{in} \quad \Omega \\ \Phi_1 &= V_0^{old} && \text{on} \quad \partial\Omega_{el}^{up} \\ \Phi_1 &= 0 && \text{on} \quad \partial\Omega^{bottom} \\ \sigma^{new}\nabla\Phi_1 \cdot n &= 0 && \text{on} \quad \partial\Omega \setminus \{\partial\Omega_{el}^{up} \cup \partial\Omega^{bottom}\}, \end{aligned} \tag{14}$$

where V_0^{old} and σ^{new} are the previous value of the potential on the upper part of the electrode and the new value of the electrical conductivity, respectively. The power dissipated by Φ_1 is as follows:

$$P_1 = \int_\Omega \sigma^{new} |\nabla\Phi_1|^2 \, \mathrm{d}x.$$

If $|P_1 - P_0| < 0.01$ W, we choose $\Phi^{new} = \Phi_1$ as new solution.

On the other hand, if P_1 does not fall within the tolerance, we proceed as follows. By linearity, the function $\Phi_2 = \lambda\Phi_1$ is the solution of problem (14) with Dirichlet boundary condition $\Phi_2 = \lambda V_0^{old}$ on $\partial\Omega_{el}^{up}$ (Dirichlet boundary conditions scale linearly and the Neumann one is itself homogeneous). The power dissipated in the domain by Φ_2 is as follows:

$$P_2 = \int_\Omega \sigma^{new} |\nabla\Phi_2|^2 \, \mathrm{d}x = \int_\Omega \sigma^{new} |\nabla(\lambda\Phi_1)|^2 \, \mathrm{d}x = \lambda^2 \int_\Omega \sigma^{new} |\nabla\Phi_1|^2 \, \mathrm{d}x = \lambda^2 P_1.$$

By requesting Φ_2 to satisfy constraint (13) (namely, imposing $P_2 = P_0$), we can solve for $\lambda = \sqrt{P_0/P_1}$, and the new potential will be as follows:

$$\Phi^{new} = \sqrt{\frac{P_0}{P_1}} \quad \Phi_1, \quad V_0^{new} = \sqrt{\frac{P_0}{P_1}} \quad V_0^{old}.$$

Time stepping

A time step of $\Delta t = 0.01s$ has been used for the bioheat equation and a smaller, adaptively computed time step is used for the Navier-Stokes equations. As a consequence of the different time steps employed, the two models are synchronized at the larger time step, where also the potential equation is solved. Let $t^n = n\Delta t$ be the nth time step of the bioheat equation, and let $(\mathbf{u}_h^n, p_h^n, \Phi_h^n, T_h^n)$ be the numerical solution at time t^n. We can briefly summarize the procedure to advance from t^n to t^{n+1} as follows.

1. Solve the Navier-Stokes equations in (t^n, t^{n+1}) with the adaptive time step and compute, in particular, $\mathbf{u}_h^{n+1}$.
2. Solve the potential equation for Φ_h^{n+1} with coefficient $\sigma(T_h^n)$.
3. Solve the bioheat equation for T_h^{n+1} using $\mathbf{u}_h^{n+1}$ in the transport term, Φ_h^{n+1} in the source term, $\sigma(T_h^n)$, $c(T_h^n)$ and $k(T_h^n)$ as coefficients.

Finally, the results are properly formatted using Tange,[44] and the lesion dimensions are calculated in a postprocessing step in Paraview[45] with self-developed Python-coded filters.

Remark 1. Since the stress tensor $\sigma(\mathbf{u}, p)$ is independent of the temperature, Equations (10a) and (10b) are in a one-way coupling with the rest of the system. Thus, already in our current situation, but even should the geometry change in time because of a variable pressure during the ablation (as it is the case in clinical RFA procedures), it would still be possible to solve the Navier-Stokes equation independently as a first step and then use the solution to solve Equations (10d) and (10c), potentially for a variety of different protocols. However, we chose to solve the Navier-Stokes equations alongside the potential and bioheat ones during each simulation in order to maintain a design flexible

enough to handle also more complex (ie, fully coupled) situations, such as a stress tensor of the tissue that varies with temperature.

Remark 2. This paper follows the scientific method, and its computational results are reproducible and replicable[46]: the simulation code (as Open Source together with reference data) is available upon request by emailing *lgerardo@bcamath.org*.

4.2 | In vitro experimental setup

Two in vitro sets of experiments using porcine myocardial tissue were performed in an experimental setup similar to the one presented in.[2] The porcine myocardial tissue was placed on a polymethyl methacrylate board in a bloodbath. An iron arm attached to the board kept the catheter perpendicular to the tissue and maintained the contact force around the target 10, 20, and 40 g, as measured by the catheter system. A pump attached to the iron arm generated a constant blood flow of 0.5 m/s. The indifferent electrode was placed at the bottom of the blood bath. A heater preserved the blood temperature to 37°C.

The six-hole open-irrigated TactiCath Quartz catheter by St. Jude Medical Inc. was used in the experiment. A constant power delivery ablation protocol of 20 or 35 W was followed for all the ablations and a saline flow rate of 17 mL/min was set. The experiments were designed based on common clinical practices, in which the ablation power ranges from 20 to 40 W with typical contact forces from 10 to 20 g and duration of 30 to 60 seconds, depending on the ablation site. For example, a typical ablation protocol in the posterior wall of the left atrium uses 10-g force and 30-W power for 30 seconds.

After the RF delivery, the lesion dimensions were assessed by examining their cross section. The measured variables were the depth (D), the width (W), and the depth of the maximum width (DW) as shown in Figure 4. In standard computational models that feature a sharp insertion of the catheter in the tissue, the depth and depth of maximum width are measured from the tissue surface. However, such measurement techniques have their limitations. First, existing computational models do not take into account the irregular surface of the cardiac wall, and neither do we at this stage. In addition, since lesions are assessed after physical manipulation, it is difficult to identify the actual position of the pre-ablation tissue surface. By considering these limitations, we opted to use measurements that allow us to validate our numerical results through comparable variables (see Figures 4 and 5, right).

A total of 31 ablations were performed for different target experimental setups, using three control parameters: the contact force F, the applied power P, and the time of ablation T. During the experiments, we observed that the target contact force was not always achieved due to maneuverability limitations and the effect of the blood inflow to the system.

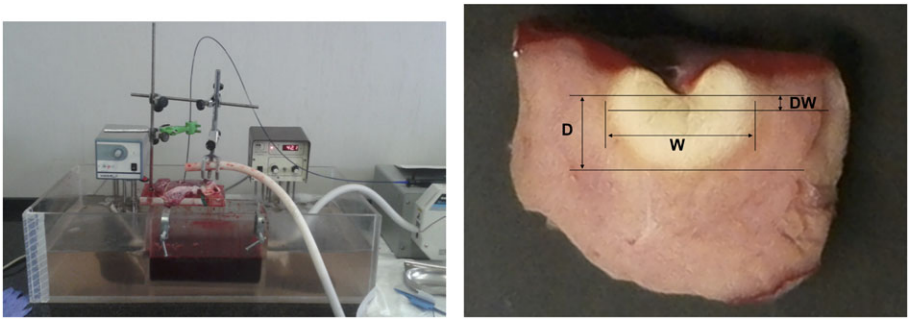


FIGURE 4 The experimental setup (left) and lesion measurements (right): depth (D), width (W), and depth of maximum width (DW)

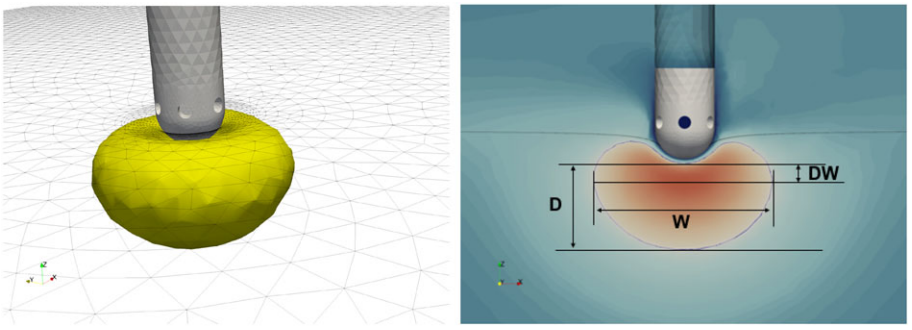


FIGURE 5 The 3D computed lesion (left) in the (20 g, 20 W, 30 s) settings and lesion measurements used for validation (right)

TABLE 2 Summary of lesion dimensions for every experimental setup. The starred values * denote the mean nominal forces and include a standard deviation of 1 g

	Target Experimental Setup (F, P, T)			
	(10^*g, 20W, 30s)	(10^*g, 35W, 30s)	(20^*g, 20W, 30s)	(40^*g, 20W, 30s)
n	8	5	9	2
$\bar{F}_{exp}$ (g)	11.2 (11.1:11.5)	11.9 (11.3:12.0)	19.9 (19.6:22.2)	41.0 (40.5:41.5)
D (mm)	2.3 (1.8:2.6)	4.4 (4.3:4.6)	3.1 (2.8:3.5)	3.3 (3.0:3.5)
min	1.4	4.2	2.2	3.0
max	3.1	4.6	4.6	3.5
W (mm)	6.5 (6.3:7.4)	12.7 (9.9:13.2)	8.9 (8.0:10.5)	7.2 (7.1:7.2)
min	5.5	8.4	7.0	7.1
max	7.9	14.1	11.7	7.2
DW (mm)	0.4 (0.4:0.7)	0.4 (-0.1:0.9)	0.5 (0.3:0.8)	0.7 (0.3:1.0)
min	0.3	−1.0	−0.6	0.3
max	0.8	1.0	1.4	1.0
R_0 (Ω)	121 (116:134)	104 (102:138)	122 (114:124)	127 (117:136)

Note. The values in the table are either absolute numbers or expressed in the format Q_2 [Q_1 : Q_3], where Q_2 is the median and Q_1 and Q_3 are the first and the third quartiles.

Thus, we allowed a standard deviation of 1 g as error of the mean experimental force F_{exp} around the nominal force F, and we removed the observations that deviate three standard deviations. Table 2 summarizes the lesion dimensions and the initial impedance of the system R_0. The mean experimental contact force $\bar{F}_{exp}$ is reported for the nonextreme cases. No complications (steam pops, char formation, and so on) were observed in our experiments.

4.3 | Model validation

We validate our model by comparing the computational lesion dimensions against the in vitro experimental results in Section 4.2. We run simulations with parameters corresponding to the first three settings of the experiments. We discarded the (40 g, 20 W, 30 s) configuration because only two experiments were available, and one of the two images obtained from the experiments was too blurry to get solid measurements. As an example, in Figure 5, we show the 3D lesion computed in the (20 g, 20 W, 30 s) settings, and the corresponding lesion dimensions used for validation. Given the uncertainty on the cross-sectional orientation (with respect to the blood flow direction) of the experimental lesions, these dimensions are assessed on the plane featuring the maximum width of the lesion.

In Table 3, we collect the resulting lesion dimensions of our numerical simulations. We observe that the depth (D), the width (W), and the depth of the maximum width (DW) of the computational lesion are within the range of the corresponding experimental values for the case of $F = 20g$. In the 10 g cases, the lesion width is underestimated by around 10% for both cases. Furthermore, the depth of the maximum width is overestimated by nearly 50% for the case of $P = 35W$. The maximum width of the lesion (W) in combination with the depth at which it occurs (DW) is an indicator used to drive the creation of continuous isolation lines in the cardiac tissue. Thus, a lack of accurate identification might yield to an imperfect isolation and the presence of residual conductivity pathways.

To dwell deeper into the width underestimation (a feature already observed in[18]), we compare the computational lesion dimensions against the interquartile range Q_1 to Q_3. The results are shown in Table 4. By comparing Tables 3 and 4, it can be seen that the depth of maximum lesion width is consistently overestimated with respect to the experimental interquartile range, while the lesion width is underestimated. The (10 g, 35 W, 30 s) setting shows a tiny deviation from the experimental lesion depth; however, the experimental results show a very narrow interquartile range of 0.3 mm. For the other settings, the depth of the lesion lies in the upper part the interquartile range or is overestimated by 5%. Nonetheless, the lesion width is always underestimated while the depth of the maximum width is overestimated up to 52.22%. This shows that the computational lesion is more spherical than the experimental one. One possible explanation for this is that our hypothesis of isotropic heat diffusion in the tissue is not good enough for the type of prediction we are trying to make. After all, it is a known fact that cardiac tissue is not isotropic itself, being made of layers of fibers that run parallel to the surface of the heart and rotate clockwise by almost 120° between the endocardium (where the electrode is in contact with the tissue) and the epicardium.[47,48] It is thus worth investigating further the anisotropy of the tissue's thermal[49] and electrical conductivities.[50] This aspect is currently under study. As the measurement of $\omega_{\max}$ in the experiments was not feasible, another possible source of DW overestimation may rely on the uncertainty about the Young's modulus of the tissue. We used a value drawn from the literature, but its effect on RFA is currently under investigation.

	Computational setup (*F*, *P*, *T*)		
	(10*g*, 20*W*, 30*s*)	**(10*g*, 35*W*, 30*s*)**	**(20*g*, 20*W*, 30*s*)**
D (mm)	2.58	4.25	3.69
min – max	✓	✓	✓
W (mm)	4.88	7.63	7.25
min – max	−11.27%	−9.17%	✓
DW (mm)	0.80	1.47	1.01
min – max	✓	+47%	✓

TABLE 3 Comparison of the computational lesions against the range of the experimental data

	Computational setup (*F, P, T*)		
	(10 *g*, 20 *W*, 30 *s*)	**(10 *g*, 35 *W*, 30 *s*)**	**(20 *g*, 20 *W*, 30 *s*)**
D (mm)	2.58	4.25	3.69
$Q_1 - Q_3$	✓	−1.16%	+5.42%
W (mm)	4.88	7.63	7.25
$Q_1 - Q_3$	−22.53%	−22.93%	−9.37%
DW (mm)	0.80	1.47	1.01
$Q_1 - Q_3$	+14.28%	+52.22%	+26.25%

TABLE 4 Comparison of the computational lesions against the interquartile range Q_1 to Q_3 of the experimental data

4.4 | Comparison of elastic and sharp insertions

One of the strengths (and novelties) of the proposed methodology resides in resorting to an elastic model for the tissue, which allows us to treat in a systematic manner both the deformation of the tissue and the power dissipated in it. The more accurate description of the tissue-electrode contact surface provided by the elastic model reflects into a more precise evaluation of the power dissipated into the tissue with respect to models featuring a sharp insertion of the catheter into an undeformed tissue. The latter is the common configuration encountered in the available literature: besides being pretty unphysical in itself, it relies on somewhat arbitrary levels of insertion in the tissue, which are never clearly explained. In addition, a sharp insertion configuration features a larger contact surface between the tissue and the electrode. Since the power delivered to the tissue depends on the percentage of electrode surface in contact with the tissue, a sharp insertion is very likely to overestimate the tissue temperature rise and, in consequence, the lesion size. To highlight this aspect, we considered two different computational geometries: an elastic deformed tissue and a sharp insertion one. In the two geometries, the catheter is placed at the same depth with respect to the undeformed tissue state. Figure 6 shows the two different computational geometries for 10, 20, and 40 g pressure. Note that in the 40 g case, the irrigation holes are submerged within the tissue. In our numerical simulations, the cooling Dirichlet boundary condition $T = 22°\text{C}$ (as described in Figure 3) is applied on the interface between the irrigation holes and the tissue. However, the holes are obstructed by the tissue, and thus, there is no saline irrigation in the blood subdomain Ω_{blood}.

We simulate an ablation protocol of 20 W for 30 seconds, where we use the tissue parameters of a porcine myocardium and we consider, inspired from existing experimental studies,[2] two different protocols with respect to the imposed blood flow velocity: an HF protocol, with $u_b = 0.5$ m/s and an LF protocol, with $u_b = 0.1$ m/s. In Figure 7, we show the impact of the two protocols on the streamlines of the saline coolant stemming from the irrigation holes in the elastic 20 g pressure case.

In Tables 5 and 6, we report the percentage α of power delivered to the tissue as well as the resulting computational lesion characteristics for the two insertion profiles in the high blood flow and low blood flow protocols, respectively. The tissue power percentage is calculated as described in Section 3.1.1. Because of the difference in the contact surface area of the electrode with the tissue, the power delivered to the tissue in the sharp insertion case is more than twice the one of the elastic case. Moreover, as this ratio grows with the insertion depth (2.15, 2.33, and 2.41 times for the 10, 20, and 40 g cases, respectively), the tissue temperature rise is expected to be more rapid and overestimated. The lesion depth is assessed as in the previous section. Concerning lesion width and depth of maximal width, we split the measurement in two: one along the direction of the blood flow (W_x at depth DW_x) and the other orthogonal to it (W_y at depth DW_y). This choice allows us to assess the difference in lesion anisotropy between the two cases. Finally, the lesion volume and surface area are assessed computing the volume enclosed by the 50°C isothermal surface in the tissue, and the area intercepted on the tissue surface by the same isothermal surface.

From Tables 5 and 6, we can observe that the sharp insertion geometry produces consistently larger maximum temperatures and steam pops, as expected from the bigger percentage of power delivered to the tissue. In the case of elastic

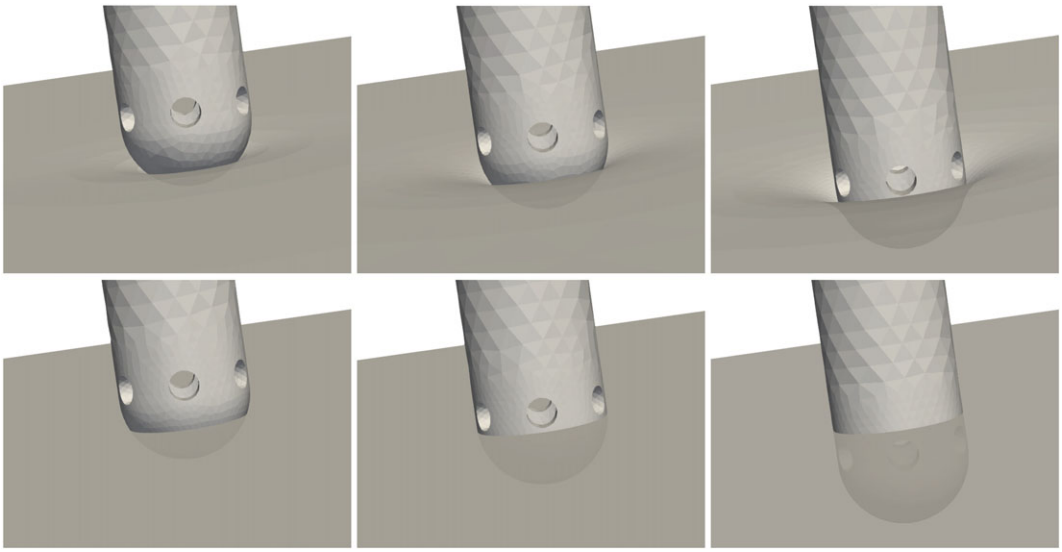


FIGURE 6 Elastic (top row) and sharp (bottom row) catheter insertion for (left) 10-g, (middle) 20-g, and (right) 40-g pressure

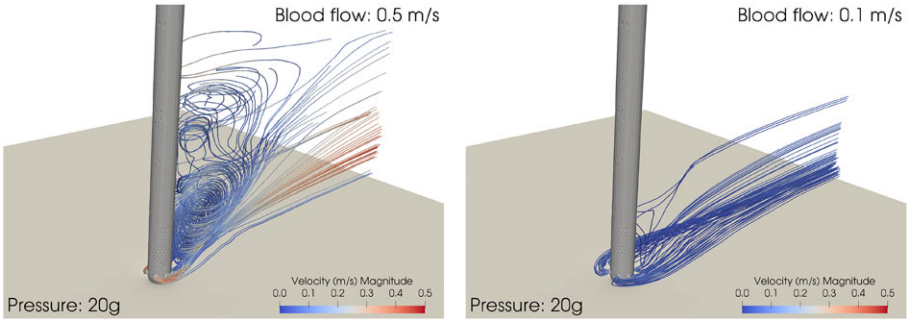


FIGURE 7 High-flow and low-flow protocols. Streamlines of the saline flow from the irrigation holes for the (left) high-flow and (right) low-flow protocols for a catheter pressure of 20 g

TABLE 5 High-flow protocol: lesion characteristics for sharp and elastic insertion

20 W, 30 s	10 g		20 g		40 g	
	Elastic	**Sharp***	**Elastic**	**Sharp***	**Elastic**	**Sharp***
α (%)	8.46	18.87	13.29	30.73	19.91	54.57
D (mm)	2.58	4.16	3.69	2.61	4.39	2.07
W_x (mm)	4.79	7.67	7.00	6.61	9.11	6.62
W_y (mm)	4.84	7.88	7.22	6.71	9.11	6.57
DW_x (mm)	0.80	1.52	0.91	0.11	0.93	-1.12
DW_y (mm)	0.80	1.32	1.01	0.21	0.93	-1.12
V (mm^3)	38.28	162.6	123.55	88.33	260.15	92.44
S (mm^2)	0	0	0	0	0	0.63
T_{max} (°C)	61.22	>100	76.95	>100	93.17	>100
Pop time (s)	–	21.6	–	5.4	–	2.5

*Simulation stopped as a pop: lesion characteristics were assessed at pop time.

10 g, the maximum tissue temperature is approximately 60°C for both protocols. On the contrary, a maximum temperature of 100°C is reached for the sharp cases after 21.6 seconds, hinting for the occurrence of steam pops. For both blood flow protocols, the lesion size before popping is much larger than the one developed using an elastic deformation of the tissue. In the 20 and 40 g cases, the sharp insertion cases have a very rapid temperature increase to 100°C in only 5.4 and 2.5 seconds respectively, hinting for the occurrence of steam pops. However, a lesion is formed using the elastic tissue and the temperature is below 100°C after the completion of 30 seconds. This comes in good agreement with our in vitro experimental results, where no steam pops were observed in the 20 and 40 g cases (see Section 4.2). In Figures 8 and 9, we compare, for both HF and LF protocols, the lesions obtained at the end of the simulation (or at the time of popping). The clipping plane for both figures is the diagonal of the computational domain. The temperature scale is the same for all plots to help visualization. The black spots in the sharp insertion cases highlight the pop location.

20 W, 30 s	10 g		20 g		40 g	
	Elastic	Sharp*	Elastic	Sharp*	Elastic	Sharp*
α (%)	8.46	18.87	13.29	30.73	19.91	54.57
D (mm)	2.46	4.17	3.62	2.63	4.38	2.09
W_x (mm)	4.44	7.74	6.69	6.77	9.08	6.76
W_y (mm)	4.41	7.91	6.86	6.83	9.21	6.70
DW_x (mm)	0.68	0.78	0.84	0.04	0.91	-1.50
DW_y (mm)	1.03	1.33	1.04	-0.26	0.91	-1.20
V (mm^3)	31.28	166.33	107.14	94.20	263.54	101.94
S (mm^2)	0	0	0	0.07	0	11.84
T_{max} (°C)	60.43	>100	75.66	>100	92.64	>100
Pop time (s)	–	21.7	–	5.3	–	2.4

*Simulation stopped as a pop occurred: lesion characteristics were assessed at pop time.

TABLE 6 Low-flow protocol: lesion characteristics for sharp and elastic insertion

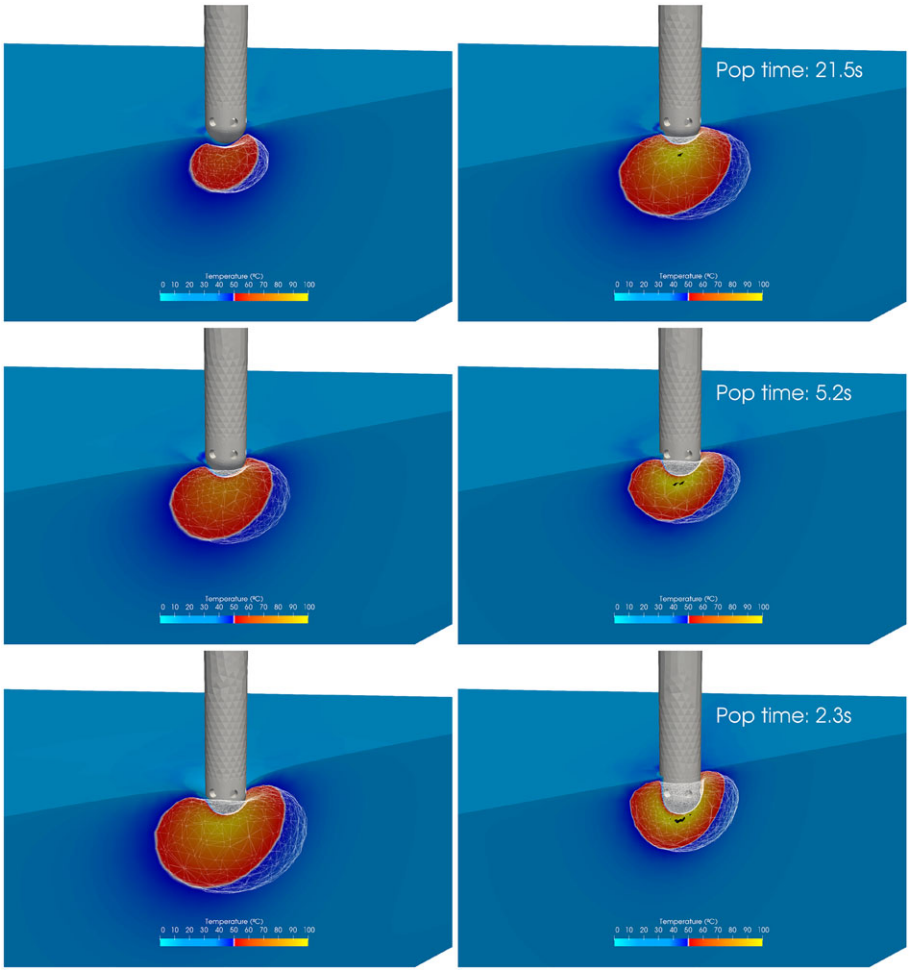


FIGURE 8 High-flow protocol. Tissue heating with (left column) elastic and (right column) sharp catheter insertions for different values of pressure. The 3D lesions are highlighted by the white wireframes. The black spots in the right panels highlight the steam pop location: top row: 10 g; middle row: 20 g; and bottom row: 40 g. The temperature scale is the same for all plots. Simulation protocol: 20 W, 30 s

Finally, we explore the impact of the high and low blood flow protocols on the elastic cases. It appears that smaller lesions are created in the LF protocol for 10 g force, while for stronger catheter contact forces such as 40 g the lesion size is similar for both protocols. This effect is a consequence of a larger portion of the catheter being in direct contact with the tissue and the fact that the lesion is formed deeper; thus, temperature changes in blood-tissue interface have a negligible effect in the lesion size. For further investigation, we consider the lesion width along the x (along the blood flow) and y (perpendicular to the blood flow) directions to evaluate the lesion morphological changes. The HF protocol produces lesions that are more symmetric, with the depth of the maximum width in the medial and lateral direction with respect to the flow being the same. However, differences in the symmetry of the lesion occur for the LF case. Figure 10

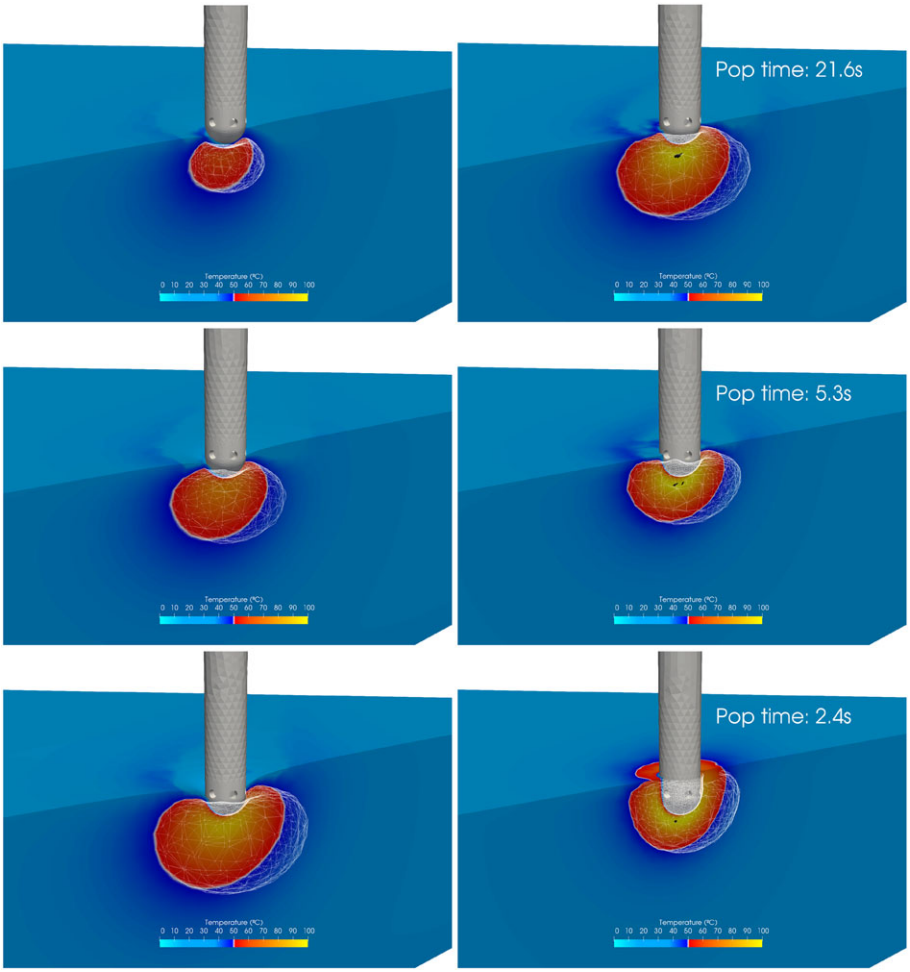


FIGURE 9 High flow vs low flow. The effect of the (top) high-flow and (bottom) low-flow protocol in the (left) medial and (right) lateral sections for the elastic catheter insertion case. The computational lesion is displayed in black outline. Simulation protocol: 10 g, 20 W, and 30 s

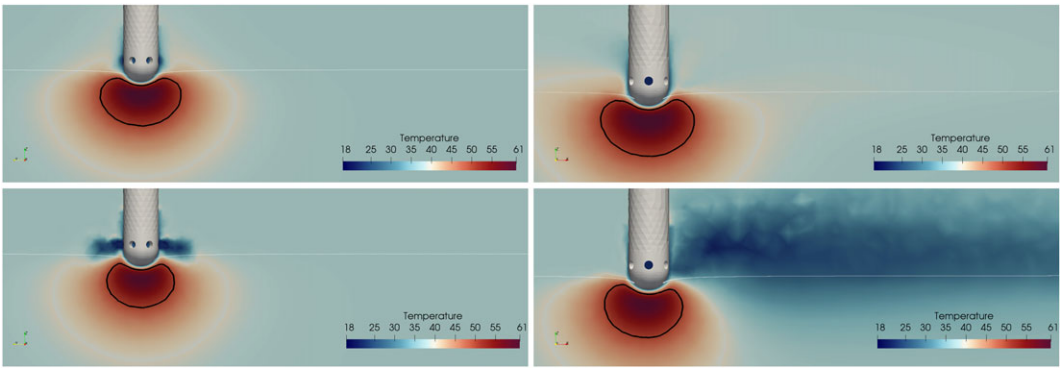


FIGURE 10 High flow vs low flow. The effect of the (top) high-flow and (bottom) low-flow protocol in the (left) medial and (right) lateral for the elastic catheter insertion case. The computational lesion is displayed in black outline. Simulation protocol: 10 g, 20 W, and 30 s

shows the temperature distribution for high and low blood flows for the elastic deformed tissue case and 10 g contact force. Note that the saline cooling effect dominates in the LF protocol and leads to a different lesion morphology near the blood-tissue interface. In particular, a small change appears in the width of the lesion in the medial direction with respect to the flow, without affecting the symmetry of the lesion shape. A more evident change in the shape of the lesion appears in the lateral section, where the lesion appears to be tilted opposite to the blood flow because of the strong cooling effect of the irrigated saline. This effect becomes less evident for higher contact force profiles, such as 20 and 40 g, and the lesion symmetry is restored in both the medial and lateral directions.

5 | DISCUSSION AND FUTURE WORK

In this paper, we propose a novel framework to model the RFCA process, which aims at surpassing important limitations of state of the art models. First, our approach does not only consider the biophysical but also the mechanical properties of the cardiac tissue. The model captures the elastic deformation that is produced on the tissue when the catheter exerts a pressure on the cardiac wall. An accurate modeling of the catheter geometry allows to account for a more realistic interaction between the coolant liquid springing from the open-irrigated catheter and the blood flow. The addition of a board between the cardiac tissue and the dispersive electrode has a double motivation. On the one hand, neither in in vitro experiments nor in clinical settings the dispersive electrode is placed in direct contact with the ablated tissue: assuming a direct contact would therefore lead to an excessive simplification of the model. On the other hand, the presence of the board allows us to use as much information as possible from the RFA system: by properly tuning the board conductivity, we can match both the appropriate power delivered to the tissue and the overall system impedance, without the need to modify the tissue conductivity. Finally, the amount of power dissipated in the tissue is computed as a function of the surface of contact between the electrode and the tissue itself. The computational model relies on open-source software, from geometry construction to numerical simulations, and postprocessing. In particular, Salome (*https://www.salome-platform.org*) was used for mesh generation, FEniCS-HPC (*https://www.fenics-hpc.org*) for the numerical solution of the PDE system and Paraview (*https://www.paraview.org*) for the postprocessing of the numerical results and the visualizations.

The computational model has been validated in comparison with two sets of *in-vitro* experiments, featuring three different ablation protocols: (10 g, 20 W, and 30 s), (10 g, 35 W, and 30 s), and (20 g, 20 W, and 30 s). The numerical lesions have been assessed against the experimental ones through some characteristic dimensions such as lesion depth, width, and depth of maximum width. The simulated lesions are globally in good agreement with the experimental measurements. In particular, the lesion depth always lies within the range of the corresponding experimental results, but the lesion width can be underestimated. A significant difference is observed in the 10 g ablation simulation, where the lesion width is underestimated by around 10%, and in the depth of the maximum width for the 35 W case, which is overestimated by nearly 50%. A more careful comparison with the interquartile Q_1 to Q_3 of the experimental results (see Table 4) shows that the depth of maximum lesion width is overestimated while the lesion width is underestimated. The simulated lesion is thus consistently more spherical than the experimental one, prompting for further investigation to clarify this issue. This shortcoming can be a consequence of the isotropic heat diffusion coefficient we used in this paper: given the lack of isotropy in the cardiac tissue itself, we are currently investigating possible anisotropies in the model parameters, in particular the thermal[49] and electrical conductivities.[50]

We then compared the effect of using an elastic deformation of the tissue under the catheter pressure against profiles featuring a sharp insertion. Accounting for tissue elasticity provides a realistic relation between the contact force and the indentation depth in the cardiac tissue. As a consequence, the actual power dissipated in the tissue can be computed in a systematic manner, differently from other state-of-the-art models featuring sharp-insertion configurations, where it is unclear how the same information is deduced. We compared the two insertion profiles by placing the catheter at the same depth with respect to the undeformed tissue. The larger amount of tissue surface in contact with the electrode in the sharp insertion case leads to a systematic overestimation of the temperature rise in the tissue. To highlight this aspect, we simulated an ablation of 20 W for 30 seconds on a porcine myocardium, with contact forces of 10, 20, and 40 g, and two different blood flow protocols. The sharp insertion cases consistently reach 100°C in less than 30 seconds, which are unphysical taken into account our in vitro experimental results, where no steam pops occurred. Furthermore, in the 40 g case, the saline irrigation holes are submerged in the cardiac tissue for a sharp profile, contrary to what has been observed during our in vitro experiments. A visual inspection on the simulated lesions shows that the elastic deformation case produces lesions that are closer in shape to the ones obtained in the experiments.

Finally, we compared the effect of the high and low blood flow protocols on the resulting lesions in the elastic case. It appears that the LF protocol produces smaller lesions for contact forces of 10 and 20 g, while the lesion size in both protocols are comparable for a contact force of 40 g in both protocols. In addition, we explored the morphological changes of the lesion between the two types of blood flows. For a contact force of 10 g, the LF protocol affects the lesion orientation, which becomes tilted opposite to the blood flow in the lateral direction because of the cooling of the irrigated saline. This effect becomes less evident as the contact force increases, and for 40 g contact force, the lesion symmetric shape is practically restored.

As accurate predictions of the lesions generated by RF cardiac ablation are fundamental to devise possible new interventional strategies, the use of elastic deformation for the tissue presented in this paper is a significant step toward a more realistic description. However, a number of limitations are still present. First, we considered an isotropic and

homogeneous cardiac tissue, even though the myocardium structure is highly complex and its surface is not smooth.[51] In addition, the deformations are assumed to lie within the elastic bound of the tissue; however, it is evident in the experimental lesions that an elasto-plastic contact occurs. This might be a result of the temperature change, which alters the mechanical properties of the tissue, an aspect worth investigating. Finally, in this work, we considered a constant Young's modulus of elasticity, even though a dependency on the heart cycle (systole or diastole) has been observed.[52] The effect of the Young's modulus of elasticity in the proposed RFA model is currently under investigation.

ACKNOWLEDGMENTS

This research was supported by the Basque Government through the BERC 2018-2021 program, by the Spanish Ministry of Economy and Competitiveness MINECO through projects MTM2015-69992-R and MTM2016-76016-R, and by the Spanish State Research Agency through BCAM Severo Ochoa excellence accreditation SEV-2017-0718 and project RTI2018-093416-B-I00 funded by (AEI/FEDER, UE) and acronym "MULTIQUANT." JJ acknowledges support from project EU H2020MSO4SC. ML acknowledges the "LaCaixa 2016" PhD grant. The experimental protocol was supported by Abbott through a nonconditional grant to JG. We acknowledge Esther Jorge-Vizuete for her valuable help during the in vitro experiments.

ORCID

Argyrios Petras https://orcid.org/0000-0002-3278-620X
Massimiliano Leoni https://orcid.org/0000-0001-5572-5234
Jose M. Guerra https://orcid.org/0000-0001-5397-9177

Johan Jansson https://orcid.org/0000-0002-1695-8809
Luca Gerardo-Giorda https://orcid.org/0000-0001-8467-1247

REFERENCES

1. Huang SKS, Wood MA. *Catheter Ablation of Cardiac Arrhythmias E-book*: Elsevier Health Sciences; 2014.
2. Guerra JM, Jorge E, Raga S, et al. Effects of open-irrigated radiofrequency ablation catheter design on lesion formation and complications: in vitro comparison of 6 different devices. *Journal of Cardiovascular Electrophysiology*. 2013;24(10):1157–1162.
3. Demolin JM, Eick OJ, Munch K, Koullick E, Nakagawa H, Wittkampf FHM. Soft thrombus formation in radiofrequency catheter ablation. *Pacing and Clinical Electrophysiology*. 2002;25(8):1219–1222.
4. Michaud GF, Kumar S. Eliminating coagulum formation with charge delivery during radiofrequency ablation: negative may be a positive!: JACC: Clinical Electrophysiology; 2016.
5. Haines DE, Verow AF. Observation on electrode-tissue interface temperature and effect on electrical impedance during radiofrequency ablation of ventricular myocardium. *Circulation*. 1990;82 (3):1034–1038.
6. Koruth JS, Dukkipati S, Gangireddy S, et al. Occurrence of steam pops during irrigated RF ablation: novel insights from Microwave Radiometry. *Journal of Cardiovascular Electrophysiology*. 2013;24(11):1271–1277.
7. Chik WWB, Kosobrodov R, Bhaskaran A, et al. Acoustic signal emission monitoring as a novel method to predict steam pops during radiofrequency ablation: preliminary observations. *Journal of Cardiovascular Electrophysiology*. 2015;26(4):440–447.
8. Labonte S. Numerical model for radio-frequency ablation of the endocardium and its experimental validation. *IEEE Transactions on Biomedical Engineering*. 1994;41(2):108–115.
9. Shimko N, Savard P, Shah K. Radio frequency perforation of cardiac tissue: modelling and experimental results. *Medical and Biological Engineering and Computing*. 2000;38(5):575–582.
10. Tungjitkusolmun S, Woo EJ, Cao H, et al. Thermal-electrical finite element modelling for radio frequency cardiac ablation: effects of changes in myocardial properties. *Medical and Biological Engineering and Computing*. 2000;38(5):562–568.
11. Berjano EJ, Hornero F. Thermal-electrical modeling for epicardial atrial radiofrequency ablation. *IEEE transactions on biomedical engineering*. 2004;51(8):1348–1357.
12. Irastorza RM, d'Avila A, Berjano E. Thermal latency adds to lesion depth after application of high-power short-duration radiofrequency energy: results of a computer-modeling study. *Journal of cardiovascular electrophysiology*. 2018;29(2):322–327.
13. González-Suárez A, Pérez JJ, Berjano E. Should fluid dynamics be included in computer models of RF cardiac ablation by irrigated-tip electrodes? *Biomedical engineering online*. 2018;17(1):43.
14. Panescu D, Whayne JG, Fleischman SD, Mirotznik MS, Swanson DK, Webster JG. Three-dimensional finite element analysis of current density and temperature distributions during radio-frequency ablation. *IEEE Transactions on Biomedical Engineering*. 1995;42(9):879–890.
15. Berjano EJ. Theoretical modeling for radiofrequency ablation: state-of-the-art and challenges for the future. *Biomedical engineering online*. 2006;5(1):24.

16. Gallagher NP, Fear EC, Vigmond EJ, Byrd IA. Cathether contact geometry affects lesion formation in radio-frequency cardiac catheter ablation. *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society, EMBS*. 2011;8(9):243–246.
17. Gopalakrishnan J. A mathematical model for irrigated epicardial radiofrequency ablation. *Annals of biomedical engineering*. 2002;30(7):884–893.
18. González-Suárez A, Berjano E, Guerra JM, Gerardo-Giorda L. Computational modeling of open-irrigated electrodes for radiofrequency cardiac ablation including blood motion-saline flow interaction. *PloS one*. 2016;11(3):e0150356.
19. Wittkampf FHM, Nakagawa H. RF catheter ablation: lessons on lesions. *Pacing and Clinical Electrophysiology*. 2006;29(11):1285–1297.
20. Cao H, Speidel MA, Tsai J-Z, Van Lysel MS, Vorperian VR, Webster JG. FEM analysis of predicting electrode-myocardium contact from RF cardiac catheter ablation system impedance. *IEEE Transactions on Biomedical Engineering*. 2002;49(6):520–526.
21. Jiang Y, Possebon R, Mulier S, et al. A methodology for constraining power in finite element modeling of radiofrequency ablation. *International Journal for Numerical Methods in Biomedical Engineering*. 2017;33(7).
22. Pérez JJ, González-Suárez A, Berjano E. Numerical analysis of thermal impact of intramyocardial capillary blood flow during radiofrequency cardiac ablation. *International Journal of Hyperthermia*. 2017;just-accepted:1–25.
23. Asaro R, Lubarda V. *Mechanics of Solids and Materials*: Cambridge University Press; 2006.
24. Popov VL. *Contact Mechanics and Friction*: Springer; 2010.
25. Yoffe E. Modified hertz theory for spherical indentation. *Philosophical Magazine A*. 1984;50(6):813–828.
26. Sneddon IN. The relation between load and penetration in the axisymmetric boussinesq problem for a punch of arbitrary profile. *International journal of engineering science*. 1965;3(1):47–57.
27. Hasgall PA, Neufeld E, Gosselin MC, Klingenböck A, Kuster N. ITIS database for thermal and electromagnetic parameters of biological tissues. *Version 3.0*. 2015.
28. Tsai J-Z, Will JA, Hubbard-Van Stelle S, et al. In-vivo measurement of swine myocardial resistivity. *IEEE Transactions on Biomedical Engineering*. 2002;49(5):472–483.
29. Duck FA. *Physical properties of tissues: a comprehensive reference book*: Academic press; 2013.
30. Bhavaraju NC, Valvano J. Thermophysical properties of swine myocardium. *International journal of thermophysics*. 1999;20(2):665–676.
31. Urban MW, Pislaru C, Nenadic IZ, Kinnick RR, Greenleaf JF. Measurement of viscoelastic properties of in vivo swine myocardium using lamb wave dispersion ultrasound vibrometry (LDUV). *IEEE transactions on medical imaging*. 2013;32(2):247–261.
32. Ramadan S, Paul N, Naguib HE. Standardized static and dynamic evaluation of myocardial tissue properties. *Biomedical Materials*. 2017;12(2):025013.
33. Nédélec J-C. *Acoustic and electromagnetic equations: integral representations for harmonic problems*: Springer Science & Business Media; 2001.
34. Jones E, Oliphant T, Peterson P. SciPy: open source scientific tools for Python. *URL https://www. scipy. org*. 2001.
35. Ribes A, Caremoli C. Salome platform component model for numerical simulation. In: Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International IEEE; 2007:553–564.
36. Geuzaine C, Remacle J-F. Gmsh: a 3-D finite element mesh generator with built-in pre-and post-processing facilities. *International journal for numerical methods in engineering*. 2009;79(11):1309–1331.
37. Schlömer N, Wagner N, Li T, et al. nschloe/meshio v1.11.7; 2018.
38. Hoffman J, Jansson J, Jansson N, De Abreu RV, Johnson C. Computability and adaptivity in CFD. *Encyclopedia of Computational Mechanics Second Edition*. 2017:1–22.
39. Logg A, Mardal K-A, Wells G, et al. Automated solution of differential equations by the finite element method. *Lecture Notes in Computational Science and Engineering*. 2012;84:1–736.
40. Hoffman J, Jansson J, Jansson N. FEniCS-HPC: automated predictive high-performance finite element computing with applications in aerodynamics. *Proceedings of the 11th International Conference on Parallel Processing and Applied Mathematics, PPAM 2015. Lecture Notes in Computer Science*. 2015.
41. Hoffman J, Jansson J, de Abreu RV, et al. Unicorn: parallel adaptive finite element simulation of turbulent flow and fluid–structure interaction for deforming domains and complex geometry. *Computers & Fluids*. 2013;80:310–319.
42. Jansson J, Krishnasamy E, Leoni M, Jansson N, Hoffman J. Time-resolved adaptive direct FEM simulation of high-lift aircraft configurations,; 2018:67–92.
43. Quarteroni AM, Valli A. *Numerical Approximation of Partial Differential Equations*: Springer Publishing Company, Incorporated; 1994.
44. Tange O. GNU parallel 20150322 ('Hellwig'). GNU Parallel is a general parallelizer to run multiple serial command line programs in parallel without changing them.; 2015.
45. Ayachit U. *The Paraview Guide: A Parallel Visualization Application*: Kitware, Inc.; 2015.
46. National Academies of Sciences, Engineering, and Medicine. *Reproducibility and replicability in science*. Washington, DC: The National Academies Press; 2019.
47. Streeter D. Gross morphology and fiber geometry in the heart. *Handbook of Physiology, Section 2: The Cardiovascular System*, Vol. 4: American Physiology Soc.; 1979:61–112.
48. Le Grice J, Smaill BH, Chai LZ, Edgar SG, Gavin JB, Hunter PJ. Laminar structure of the heart: ventricular myocyte arrangement and connective tissue architecture in the dog. *Am. J. Physiol.* 1995;269 (2 Pt. 2):H571–H582.
49. Bhattacharya A, Mahajan R. Temperature dependence of thermal conductivity of biological tissues. *Physiological measurement*. 2003;24(3):769.
50. Xie F, Zemlin CW. Effect of twisted fiber anisotropy in cardiac tissue on ablation with pulsed electric fields. *PloS one*. 2016;11(4):e0152262.

51. Ho SY, Sanchez-Quintana D, Cabrera JA, Anderson RH. Anatomy of the left atrium: implications for radiofrequency ablation of atrial fibrillation. *Journal of cardiovascular electrophysiology*. 1999;10(11):1525–1533.
52. Couade M, Pernot M, Tanter M, et al. Quantitative imaging of myocardium elasticity using supersonic shear imaging. In: Ultrasonics Symposium (IUS), 2009 IEEE International IEEE; 2009:151–154.

How to cite this article: Petras A, Leoni M, Guerra J, M, Jansson J, Gerardo-Giorda L. A computational model of open-irrigated radiofrequency catheter ablation accounting for mechanical properties of the cardiac tissue. *Int J Numer Meth Biomed Engng*. 2019;e3232. https://doi.org/10.1002/cnm.3232

Tissue Drives Lesion: Computational Evidence of Interspecies Variability in Cardiac Radiofrequency Ablation

Argyrios Petras[1(✉)], Massimiliano Leoni[1,2], Jose M. Guerra[3], Johan Jansson[1,2], and Luca Gerardo-Giorda[1]

[1] BCAM - Basque Center for Applied Mathematics, 48009 Bilbao, Spain
apetras@bcamath.org
[2] KTH Royal Institute of Technology, 11428 Stockholm, Sweden
[3] Hospital de la Santa Creu i San Pau, CIBERCV, 08041 Barcelona, Spain

Abstract. Radiofrequency catheter ablation (RFCA) is widely used for the treatment of various types of cardiac arrhythmias. Typically, the efficacy and the safety of the ablation protocols used in the clinics are derived from tests carried out on animal specimens, including swines. However, these experimental findings cannot be immediately translated to clinical practice on human patients, due to the difference in the physical properties of the types of tissue. Computational models can assist in the quantification of this variability and can provide insights in the results of the RFCA for different species. In this work, we consider a standard ablation protocol of 10 g force, 30 W power for 30 s. We simulate its application on a porcine cardiac tissue, a human ventricle and a human atrium. Using a recently developed computational model that accounts for the mechanical properties of the tissue, we explore the onset and the growth of the lesion along time by tracking its depth and width, and we compare the lesion size and dimensions at the end of the ablation.

Keywords: Radiofrequency catheter ablation · Mathematical model · Tissue properties · Interspecies variability

1 Introduction

Radiofrequency ablation (RFA) is a common treatment for cardiac arrhythmias. Through a catheter advanced into the patient's heart, radiofrequency current is delivered to the tissue which produces resistive heating in the neighborhood of the electrode, while conduction propagates the heat to the immediate surrounding tissue. At a temperature of 50 °C the tissue is irreversibly damaged and a permanent lesion develops [3]. RFA is generally safe, however life-threatening

Supported by the Basque Government through BERC 2019–2021 and by the Spanish Ministry of Economy and Competitiveness MINECO through BCAM Severo Ochoa excellence accreditation SEV-2017-0718.

Y. Coudière et al. (Eds.): FIMH 2019, LNCS 11504, pp. 139–146, 2019.
https://doi.org/10.1007/978-3-030-21949-9_16

complications can occur, including the formation of thrombi due to blood overheating at 80 °C and steam pops at temperatures of 100 °C within the tissue, which are among the most severe complications [3].

Typically, ablation protocols are designed and tested using *in-vitro*, *ex-vivo* and *in-vivo* experiments on animals. Porcine cardiac tissue is commonly used in the experiments to assess the efficacy and the safety of the RFA treatment [1,5]. However, the biophysical, mechanical and physiological properties of the porcine cardiac tissue differ from the corresponding human ones [6], thus a direct translation of the experimental results to clinical practice can lead to insufficient treatment or potential life-threatening complications. Computational models can be a valuable asset in the assessment of RFA protocols. The efficacy and safety of the procedure can be directly assessed on simulated human tissue using the reported physical properties in the literature, thus avoiding the translation of data from experiments that use porcine tissue.

In this study, we investigate the interspecies variability in the RFA treatment using a recently developed computational model [5] which includes the mechanical properties of the tissue. We simulate a commonly used ablation protocol on cardiac tissue of two species: porcine and human. Two different ablation sites are considered in the case of the simulated human tissue: an atrium and a ventricle.

2 Computational Model

2.1 Geometry

Based on an experimental setup similar to [1], an 80 mm × 80 mm × 80 mm box is constructed that includes blood (top 40 mm), cardiac tissue of thickness H mm and an external factors board (bottom $40 - H$ mm). Within the blood compartment, the catheter is placed perpendicularly to the tissue at the middle of the box. We consider a 6-hole electrode with a hemispherical tip and a thermistor inspired by RFA catheters commonly used in clinics, as described in [5]. Figure 1 shows a sample computational geometry for a tissue thickness of $H = 20$ mm.

Due to the contact with the catheter, the tissue undergoes a mechanical deformation. The vertical deformation is described in [5] and is formulated as

$$\omega(r) = \begin{cases} \omega_{\max} - (R - \sqrt{R^2 - r^2}), & r \leq a, \\ \\ \frac{a^2}{\pi} \int_0^1 \frac{2\omega_{\max} - ax \log\left(\frac{R+ax}{R-ax}\right)}{r^2 - a^2x^2}\, dx, & r > a, \end{cases}$$

where R is the radius of the electrode, $\omega_{\max}$ is the maximum indentation depth, a is the contact radius of the electrode with the tissue and r is the planar distance from the center of the electrode. The indentation depth and contact radius can be computed using the contact force F, the Young's modulus E and the Poisson's ratio ν of the tissue, as follows:

$$\omega_{\max} = \frac{a}{2} \log\left(\frac{R+a}{R-a}\right),$$

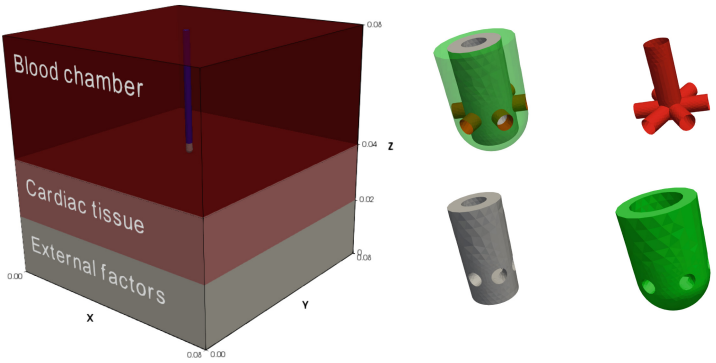


Fig. 1. Left: The full computational geometry. Right: The computational tip of the catheter (top left), the saline tubes (top right), the thermistor (bottom left) and the electrode (bottom right).

$$F = \frac{E}{2(1-\nu^2)}\left((a^2+R^2)\log\left(\frac{R+a}{R-a}\right) - 2aR\right).$$

The deformation of the tissue is directly embedded in the construction of the computational geometry.

2.2 Governing Equations

The blood flow and its interaction with the irrigated saline in the blood compartment Ω_{blood} of the geometry are modelled using the incompressible Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\,\mathbf{u} - \operatorname{div}\sigma(\mathbf{u},p) = \mathbf{0},$$
$$\operatorname{div}\mathbf{u} = 0,$$

where $\mathbf{u}$ is the velocity, $\sigma(\cdot,\cdot)$ is the stress tensor and p is the scaled pressure. A constant inflow is considered in one of the sides of Ω_{blood} with its corresponding outflow conditions at the opposite side. A constant inflow is also considered from the saline irrigation holes on the electrode flowing radially within Ω_{blood}. The remaining interfaces of Ω_{blood} are equipped with no slip conditions.

A modified Penne's bioheat equation tracks the changes in the temperature of the geometry Ω over time

$$\rho c(T)\left(\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla\,T\right) - \operatorname{div}(k(T)\,\nabla\,T) = \sigma(T)|\,\nabla\,\Phi|^2,$$

where T is the temperature, t is the time, ρ is the density, $c(T)$ is the specific heat, $\mathbf{u}$ is the velocity, $k(T)$ is the thermal conductivity, $\sigma(T)$ is the electrical conductivity and Φ is the electrical potential. A temperature of 22 °C is assumed on the interface between the irrigation holes and the blood compartment Ω_{blood}

to model the cooling effect of the saline, while insulation boundary conditions are applied on the irrigation tubes and the catheter body. A constant body temperature of 37 °C is imposed on all outer boundaries of the computational domain Ω.

The spatial distribution of the potential follows a quasi-static equation, augmented with a power constraint for constant power ablations

$$\begin{cases} \operatorname{div}(\sigma(T)\,\nabla\,\Phi) = 0, \\ \int_\Omega \sigma(T)|\nabla\Phi|^2\,dx = P, \end{cases}$$

where P is the power dissipated within the computational domain Ω. A potential V_0 is applied on the interface of the catheter and the electrode, tuned to match the power P dissipated in the computational domain Ω. The dispersive electrode is placed at the bottom of Ω, where zero potential is imposed. All the remaining surfaces are considered electrically insulated. More details on the computational model can be found in [5].

2.3 Parameters

The parameters considered in the computational model are summarized in Table 1. The specific heat c, thermal and electrical conductivities k and σ are considered temperature dependent within the tissue:

$$\begin{aligned} c(T) &= c_0(1 + c_1(T - 37)) \\ k(T) &= k_0(1 + k_1(T - 37)) \\ \sigma(T) &= \sigma_0(1 + \sigma_1(T - 37)) \end{aligned}$$

The electrical conductivity of the external effects board σ_b is tuned to match the initial resistance of the system, and the power delivered to the tissue, which is calculated as follows:

$$P_{tissue} = \frac{A_{tissue}\sigma_0^{(tissue)}}{A_{blood}\sigma_0^{(blood)} + A_{tissue}\sigma_0^{(tissue)}}\,P_{abl} =: \alpha P_{abl},$$

where A_{tissue} and A_{blood} are the surface areas of the electrode in contact with the tissue and the blood respectively, $\sigma_0^{(tissue)}$ and $\sigma_0^{(blood)}$ are the electrical conductivities of the tissue and the blood at body temperature and P_{abl} is the total power set by the ablation protocol. The initial resistance of the system is set as 120 Ω. More details on the calculation of σ_b can be found in [5].

In our extensive literature review, no work has been found that addresses differences in biophysical properties on different regions of the human heart. Thus, we assume that the human atrium and ventricle have the same biophysical properties, which are drawn from [2] at body temperature. The temperature dependence of the specific heat and thermal conductivity can be calculated from [10]. No data were found for the dependence of the electrical conductivity of the

Table 1. The summary of the biophysical and mechanical parameters that appear in the computational model.

Parameters	Blood	Electrode	Thermistor	Board	Tissue	
					Porcine	Human
ρ (kg m^3)	1050	21500	32	1076	1076	1081
c_0 (J kg^{-1} K^{-1})	3617	132	835	3017	3017	3686
c_1 ($^\circ$C^{-1})	-	-	-	-	-0.0042	-0.0011
k_0 (W m^{-1} K^{-1})	0.52	71	0.038	0.518	0.518	0.56
k_1 ($^\circ$C^{-1})	-	-	-	-	-0.0005	0.0022
σ_0 (S m^{-1})	0.748	4.6×10^6	10^{-5}	σ_b	0.54	0.381
σ_1 ($^\circ$C^{-1})	-	-	-	-	0.015	0.015
ν (-)	-	-	-	-	0.499	0.499
E (kPa)	-	-	-	-	75	40

myocardium on the temperature, thus values for the human liver are considered, since a similar behavior is reported for other species [6]. The mechanical properties for human cardiac tissue are summarized in [7,9].

The thickness of the porcine cardiac wall is considered as $H = 20$ mm [4]. Two different substrates are modelled for the simulated human cardiac tissue: an atrium of $H = 6$ mm thickness and a ventricle of $H = 12.5$ mm [8,11].

3 Results

We simulate a standard constant power ablation protocol of 30 W, with a contact force of 10 g for a total of 30 s of ablation. The blood flow protocol is set to 0.1 m s^{-1} and the saline irrigation rate to 17 ml min^{-1}. This ablation protocol is typically used in RFA experiments [1].

We identify the computational lesion by the 50 °C isotherm contour: the quantities measured are the depth (D), the width (W), the depth at which the maximum width occurs (DW), the surface area of the lesion (SA) and the volume (V). In addition to the lesion size dimensions mentioned above, we track the maximum temperature in the tissue and the blood. Details on the measurement of these quantities can be found in [5].

The computational results are shown in Table 2. The lesion on the human atrium appears to be the biggest in volume, while the smallest is on the human ventricle. The depth in the porcine cardiac tissue and the human atrium are comparable, while the lesion is more shallow in the human ventricle. A bigger difference appears in the measured width. In particular, a difference of nearly 1 mm is observed between the cases of human atrium and ventricle, while the porcine lesion width is comparable with the human ventricle one. The morphology of all three lesions appears to be different, with significant changes in the depth at which the maximum width occurs. Though the maximum width of the

Table 2. The summary of the temperature and lesion size dimensions after the completion of 30 s of ablation for the simulated tissue considered.

	Porcine	Human ventricle	Human atrium
D (mm)	3.76	3.35	3.75
W (mm)	6.73	6.56	7.42
DW (mm)	1.38	0.86	0.94
SA (mm^2)	0.11	0.0	0.0
V (mm^3)	103.4	92.3	124.9
$T_{\max}$ tissue (°C)	77.7	71.9	77.3
$T_{\max}$ blood (°C)	51.5	56.1	62.0

lesions on the human specimen appears to be more shallow, there is no surface burning. A small surface burn is present in the case of the porcine cardiac tissue.

The temperature of the tissue is similar in the cases of the porcine cardiac tissue and the human atrium, while it is about 5 °C lower in the human ventricle. On the contrary, the blood temperature is much lower in the porcine cardiac tissue. The temperature difference with respect to the human tissue is bigger than 10 °C for the atrium and around 5 °C for the ventricle.

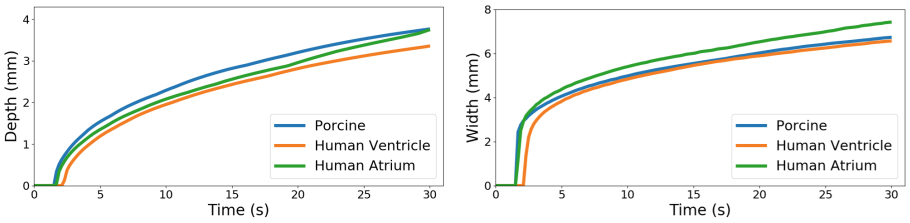


Fig. 2. The lesion depth (left) and width (right) over time for the simulated porcine cardiac tissue, human ventricle and human atrium.

To further compare the results on the three types of tissue considered, we explore in Fig. 2 the evolution of the width and the depth of the lesion throughout the duration of the ablation. We observe that the lesion is created almost at the same time on the porcine cardiac tissue and the human atrium, while the one in the human ventricle is delayed by 0.8 s. The depth of the lesion in the porcine case is consistently larger than the one in the human tissue. The depth of the lesion in the human atrium case is initially similar to the human ventricle case, but after 20 s it increases to a depth comparable to the porcine case. On the contrary, the width of the lesion in the human atrium is larger than the other cases throughout the duration of the ablation, while comparable values appear between the human ventricle and the porcine tissue.

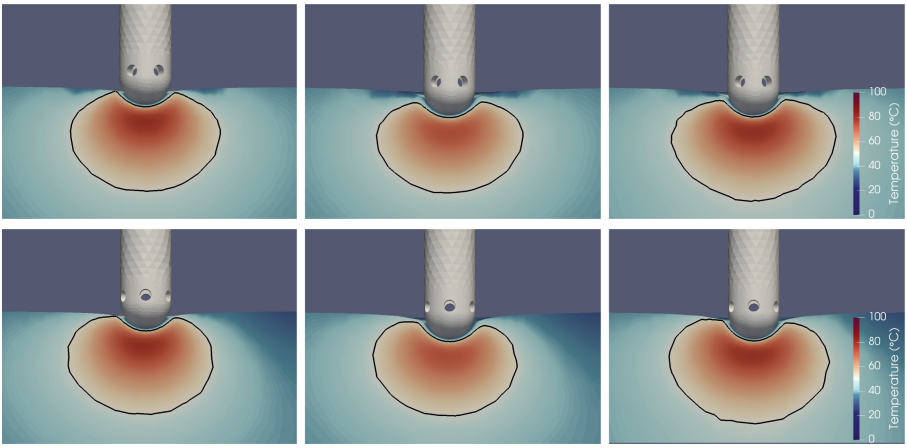


Fig. 3. The medial (top) and lateral (bottom) view of the lesion for porcine cardiac tissue (left), human ventricle (middle) and human atrium (right) at the final time of the ablation.

Finally, to explore morphological differences of the lesions, we show in Fig. 3 the medial and lateral view of the 50 °C isotherm contour within the tissue at the final time of the ablation. All three lesions appear tilted towards the direction of the blood in the lateral view, while they are symmetric in the medial view, as previously observed in [5]. The lesions in the human tissue appear elongated and less spherical than the ones in the porcine tissue, which is reflected in the difference in the depth of the maximum width in Table 2.

4 Conclusions

We explored interspecies variability during RFA treatment, comparing the resulting lesion size dimensions of the ablation process on a porcine cardiac tissue, a human ventricle and a human atrium. Our computational results indicate that the lesion characteristics are different in all three cases. The human atrium provides the largest lesion overall. The depth of the lesion in the porcine cardiac tissue case is larger than the one on the human tissue throughout the ablation process. Thus, a direct application to a human patient of a protocol based on the achievement of a specific lesion depth in porcine tissue would result in too small lesions and insufficient treatment in cases that transmural lesions are targeted.

The width of the lesion in the human atrium is at least 0.7 mm larger than the other two cases. In addition, the lesions in the simulated human tissue appear less spherical than the porcine one. A direct translation from porcine data to the human atrium would affect the efficiency of the RFA process, indicating that a larger number of lesions are required when targeting isolation lines in the atrium.

The maximum tissue temperature in the atrium and the porcine tissue are comparable, however there is a difference of more than 10 °C the maximum

blood temperature. Regarding the human ventricle, the tissue temperature is lower than the porcine one, however the maximum blood temperature is about 5 °C higher. This indicates that the results on porcine tissue can lead to blood overheating and thrombus formation if directly translated to human tissue. On the contrary, steam pops would be avoided as the maximum tissue temperature is overestimated, though the efficiency of the process can be reduced by inducing smaller lesions in the human ventricle.

References

1. Guerra, J.M., et al.: Effects of open-irrigated radiofrequency ablation catheter design on lesion formation and complications: in vitro comparison of 6 different devices. J. Cardiovasc. Electrophysiol. **24**(10), 1157–1162 (2013)
2. Hasgall, P., Neufeld, E., Gosselin, M., Klingenböck, A., Kuster, N.: ITIS database for thermal and electromagnetic parameters of biological tissues, Version 3.0 (2015)
3. Huang, S.K.S., Wood, M.A.: Catheter Ablation of Cardiac Arrhythmias E-book. Elsevier Health Sciences, Philadelphia (2014)
4. Liu, S.K., et al.: Hypertrophic cardiomyopathy in pigs: quantitative pathologic features in 55 cases. Cardiovasc. Pathol. **3**(4), 261–268 (1994)
5. Petras, A., Leoni, M., Guerra, J.M., Jansson, J., Gerardo-Giorda, L.: A computational model of open-irrigated radiofrequency catheter ablation accounting for mechanical properties of the cardiac tissue. arXiv preprint arXiv:1810.09157 (2018)
6. Rossmann, C., Haemmerich, D.: Review of temperature dependence of thermal properties, dielectric properties, and perfusion of biological tissues at hyperthermic and ablation temperatures. Crit. Rev. Biomed. Eng. **42**(6), 467 (2014)
7. Rump, J., Klatt, D., Braun, J., Warmuth, C., Sack, I.: Fractional encoding of harmonic motions in MR elastography. Magn. Reson. Med. **57**(2), 388–395 (2007)
8. Sjögren, A.L.: Left ventricular wall thickness determined by ultrasound in 100 subjects without heart disease. Chest **60**(4), 341–346 (1971)
9. Strachinaru, M., et al.: Cardiac shear wave elastography using a clinical ultrasound system. Ultrasound Med. Biol. **43**(8), 1596–1606 (2017)
10. Valvano, J.W., Cochran, J.R., Diller, K.R.: Thermal conductivity and diffusivity of biomaterials measured with self-heated thermistors. Int. J. Thermophys. **6**(3), 301–311 (1985)
11. Whitaker, J., et al.: The role of myocardial wall thickness in atrial arrhythmogenesis. Ep Eur. **18**(12), 1758–1772 (2016)

Effect of Tissue Elasticity in Cardiac Radiofrequency Catheter Ablation Models

Argyrios Petras[1], Massimiliano Leoni[1,2], Jose M Guerra[3], Johan Jansson[1,2], Luca Gerardo-Giorda[1]

[1] Basque Center for Applied Mathematics, Bilbao, Spain
[2] KTH Royal Institute of Technology, Stockholm, Sweden
[3] Hospital de la Santa Creu i San Pau, Barcelona, Spain

Abstract

Radiofrequency catheter ablation (RFCA) is an effective treatment for different types of cardiac arrhythmias. However, major complications can occur, including thrombus formation and steam pops. We present a full 3D mathematical model for the radiofrequency ablation process that uses an open-irrigated catheter and accounts for the tissue deformation, an aspect overlooked by the existing literature. An axisymmetric Boussinesq solution for spherical punch is used to model the deformation of the tissue due to the pressure of the catheter tip at the tissue-catheter contact point. We compare the effect of the tissue deformation in the RFCA model against the use of a standard sharp insertion of the catheter in the tissue that other state-of-the-art RFCA computational models use.

1. Introduction

Radiofrequency catheter ablation is a minimally invasive, effective treatment for various types of cardiac arrhythmias. Typically, for endocardiac RFCA, a catheter is inserted in the cardiac chamber and placed at the arrhythmogenic tissue. Resistive electrical heating is produced in a neighborhood of the electrode, which propagates by conduction to the rest of the tissue, causing irreversible damage to the latter and the formation of a lesion at $50\,^{\circ}\mathrm{C}$ [1]. Radiofrequency ablation (RFA) is an effective and a safe procedure for cardiac arrhythmias; however a number of life-threatening complications can occur, including the possibility of thrombus formation, in case the blood temperature rises above $80\,^{\circ}\mathrm{C}$, and steam pops in the occurrence of tissue overheating (around $100\,^{\circ}\mathrm{C}$) [1].

Several computational models have been introduced to describe the biophysics of RFA in two and three dimensions [2,3]. Typically, a modified Penne's bioheat equation is considered for the temperature changes, while a quasi-static equation describes the spatial distribution of the electrical potential. Some models include the blood flow either as boundary conditions for the bioheat equation in the tissue or using the Navier-Stokes equations. However, the state-of-the-art models overlook the mechanical deformation of the tissue due to the contact with the catheter. Until recently, only [4] used profiles extracted from X-ray scanning to identify the deformation of the tissue; however this approach is partial and dependent on the tissue specimen. We introduced in [5] a methodological approach, using a solution of the axisymmetric Boussinesq problem for a spherical punch. The model has been validated against experiments designed ad-hoc [5]. We explore here the effect of the tissue elastic deformation against the sharp insertion (undeformed case) currently used from state-of-the-art models in the RFA modelling.

2. Mathematical Model

2.1. Computational geometry

A box of size $80\,\mathrm{mm} \times 80\,\mathrm{mm} \times 80\,\mathrm{mm}$ is considered, that consists of five different components: the blood chamber $80\,\mathrm{mm} \times 80\,\mathrm{mm} \times 40\,\mathrm{mm}$, the cardiac tissue $80\,\mathrm{mm} \times 80\,\mathrm{mm} \times 20\,\mathrm{mm}$, a board $80\,\mathrm{mm} \times 80\,\mathrm{mm} \times 20\,\mathrm{mm}$ that models the external effects between the tissue and the dispersive electrode, the electrode and the thermistor. The electrode is placed at the center of the blood chamber, has $3.5\,\mathrm{mm}$ length and $2.33\,\mathrm{mm}$ diameter, contains 6 irrigation holes of diameter $0.5\,\mathrm{mm}$ and has a hemispherical tip of radius $1.165\,\mathrm{mm}$. The thermistor is placed inside the electrode and has a diameter of $1.54\,\mathrm{mm}$ and a length of $3\,\mathrm{mm}$. Figure 1 shows the geometry considered.

2.2. Governing equations

During the RFA treatment, the catheter is placed on the arrhythmogenic tissue and causes the deformation of the latter. Using an axisymmetric Boussinesq solution for a spherical indenter in contact mechanics [6], the vertical deformation of the tissue is described as

 ISSN: 2325-887X DOI: 10.22489/CinC.2018.035

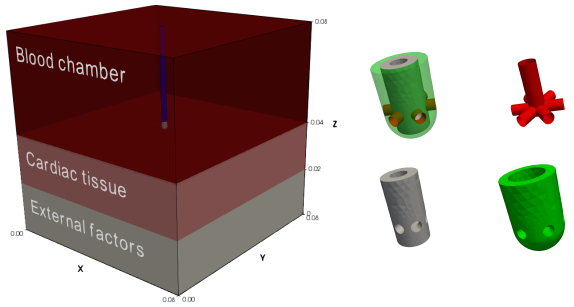


Figure 1. Left: The full computational geometry. Right: The computational tip of the catheter (top left), the saline tubes (top right), the thermistor (bottom left) and the electrode (bottom right).

$$\Delta z(r) = \begin{cases} \omega_{\max} - (R - \sqrt{R^2 - r^2}), & r \le a, \\ \dfrac{a^2}{\pi} \displaystyle\int_0^1 \dfrac{2\omega_{\max} - at \log\left(\frac{R+at}{R-at}\right)}{r^2 - a^2 t^2} dt, & r > a, \end{cases}$$

where R is the radius of the hemispherical tip of the electrode, r is the distance from the center of the electrode, $\omega_{\max}$ is the maximum deformation (which occurs at the center of the electrode) and a is the contact radius of the electrode and the tissue. The maximum depth and the contact radius can be calculated using the contact force F as

$$F = \frac{G}{1-\nu}\left((a^2 + R^2)\log\left(\frac{R+a}{R-a}\right) - 2aR\right),$$

$$\omega_{\max} = \frac{a}{2}\log\left(\frac{R+a}{R-a}\right),$$

where G is the shear modulus and ν is the Poisson's ratio of the tissue. The elastic deformation Δz is applied to the cardiac tissue in the computational geometry.

The catheter is surrounded by blood and continuously flushes saline through the irrigation holes on the electrode at its tip to cool the ablation site. The incompressible Navier-Stokes equations are employed to model the blood-saline interaction. A constant inflow is considered on a side of the blood chamber, with the corresponding outflow boundary conditions on the opposite side; a radial inflow towards the blood is considered from the saline tubes. All the remaining boundaries, including the blood-tissue interface, are equipped with no slip conditions.

During the ablation, electrical current flows through the catheter to the tissue, causing Joule heating. A modified Penne's bioheat equation is typically used to describe the temperature change [3]

$$\rho c(T)\left(\frac{\partial T}{\partial t} + \mathbf{v}\cdot\nabla T\right) - \nabla\cdot(k(T)\nabla T) = q + Q_m - Q_p,$$

where T is the temperature, $\mathbf{v}$ is the velocity from the Navier-Stokes equations, ρ is the density, c is the specific heat, k is the thermal conductivity, Q_m is the metabolic heat, Q_p is the heat loss from blood perfusion and q is the heat source. For endocardiac RFA, the metabolic and blood perfusion heat Q_m and Q_p are negligible and thus omitted in our model [3].

The heat source produced by the electrical potential is $q = \sigma(T)|\nabla\Phi|^2$, where σ is the electrical conductivity and Φ is the electrical potential. For current at frequencies of $500\,\mathrm{kHz}$ and over the distance of interest, the system is considered totally resistive, thus a quasi-static equation can be employed to model the potential distribution

$$\nabla\cdot(\sigma(T)\nabla\Phi) = 0.$$

For constant power ablation, the quasi-static equation is augmented with a constraint

$$\int_\Omega \sigma(T)|\nabla\Phi|^2 dx = P,$$

where Ω is the computational domain and P is the total power dissipated in the system.

Regarding the boundary conditions of the above equations, the body temperature of $37\,^\circ\mathrm{C}$ is applied on all outer boundaries of the computational geometry. The temperature on the irrigation holes is set to $22\,^\circ\mathrm{C}$, and insulation boundary conditions are applied on the interfaces shared with the catheter body and on the saline pipes (shown in red in Figure 1). In terms of the boundary conditions for the electrical potential, a constant voltage V_0 is imposed on the catheter-electrode interface (the top surface of the green electrode in Figure 1) calculated to satisfy the power constraint equation. Zero potential is enforced at the bottom of the computational geometry, while zero flux conditions are imposed in all the remaining boundaries.

2.3. Power distribution

The machines used in the RFA treatment provide the impedance of the full system during the ablation. In our computational model, the electrical conductivity of the external effects board (see Figure 1) is tuned to match the initial impedance provided by the RFA machine. In particular, for a constant power ablation, the initial V_0 can be identified by Ohm's law. By enforcing V_0 in the quasi-static electrical potential equation, and by enforcing a constraint

$$\int_{\Omega_t} \sigma(T)|\nabla\Phi|^2 dx = P_t,$$

where Ω_t is the cardiac tissue and P_t is the power delivered to the tissue, optimization techniques are used to identify the electrical conductivity of the board [5]. The quantity P_t depends on the tissue electrical conductivity and the tissue-electrode contact surface area, and can be calculated using the technique described by Wittkampf et al. [7]. In particular,

$$P_t = \frac{A_t \sigma_t}{A_b \sigma_b + A_t \sigma_t} P_{nom} =: \alpha P_{nom},$$

where A_t and A_b are the surface area of the electrode in contact with the tissue and the blood respectively, σ_t and σ_b are the tissue and the blood electrical conductivities and P_{nom} is the nominal power of the ablation protocol. The electrical conductivity of the board remains constant throughout the RFA simulation.

Due to the scaling and other external factors, the total power from the RFA machine cannot coincide with the power dissipated within our computational geometry. Thus, the total power dissipated in our computational system P is calculated in a preprocessing step as

$$\int_\Omega \sigma(T) |\nabla \Phi|^2 dx = P.$$

This power P is kept constant throughout the RFA simulation, and corresponds to the constraint equation imposed on the electrical potential quasi-static equation.

3. Results

In our computational simulations, the open source software Salome [8] is used for the geometry generation, FEniCS-HPC [9] for the numerical solution of our model using the finite element method and Paraview [10] for the postprocessing of the computational lesion and the visualizations.

3.1. Parameters

The parameters used in the mathematical model are collected from the literature. In particular, the physical parameters of the electrode and the thermistor can be found in [3]. The blood physical parameters are found in the virtual population database from the IT IS foundation [11]. A porcine cardiac tissue is considered with electrical properties as appear in [2] and thermal properties as described in [12]. In particular, a linear decrease of the specific heat with respect to temperature of 0.42% from the value $3017\,\mathrm{J\,kg^{-1}\,K^{-1}}$ at body temperature is considered. Similarly, for the thermal conductivity a decrease of 0.05% from the value $0.518\,\mathrm{W\,m^{-1}\,K^{-1}}$ at body temperature, and for the electrical conductivity an increase of 1.5% from the value $0.54\,\mathrm{S\,m^{-1}}$ at body temperature. Regarding the mechanical properties of the cardiac tissue, the Poisson's ratio is chosen as 0.499 and the shear modulus as $25\,\mathrm{kPa}$ [13]. Finally, the physical parameters of the external effects board are chosen to be constant, as the corresponding ones of the cardiac tissue at body temperature.

3.2. Discussion

In our numerical simulations, a constant power ablation protocol is considered of $20\,\mathrm{W}$ for $30\,\mathrm{s}$. The saline inlet is set as $17\,\mathrm{mL/\,min}$, following the instructions of the catheter producer. Three different contact forces are considered: $10\,\mathrm{g}$, $20\,\mathrm{g}$ and $40\,\mathrm{g}$. In the undeformed tissue case, the catheter is placed at the same depth as the one calculated for the deformed case. The power delivery is controlled according to the procedure described in the previous section for both the elastic and sharp insertions. Two blood flow velocity profiles are considered that are commonly used in experiments: a high blood flow of $0.5\,\mathrm{m\,s^{-1}}$ and a low blood flow of $0.1\,\mathrm{m\,s^{-1}}$. The computational lesions are identified by tracking the $50\,^\circ\mathrm{C}$ isotherm contour. Figure 2 shows the numerical results for the case of high blood flow, for different contact forces. In case of steam pop occurrence at $100\,^\circ\mathrm{C}$, the lesion is shown right at the popping time.

In the case of $10\,\mathrm{g}$ and high blood flow, a small lesion is formed in the elastic case after the completion of $30\,\mathrm{s}$, while the temperature in the sharp insertion profile case reaches $100\,^\circ\mathrm{C}$ after $21.6\,\mathrm{s}$, which hints for steam pop formation. Similarly, in the cases of $20\,\mathrm{g}$ and $40\,\mathrm{g}$, the elastic case provides a lesion with maximum tissue temperature of $77\,^\circ\mathrm{C}$ and $93\,^\circ\mathrm{C}$ respectively, while steam pops occur in the sharp case after the completion of $5.4\,\mathrm{s}$ and $2.5\,\mathrm{s}$. Similar observations can be drawn for the results of the low blood flow.

4. Conclusions

The elastic deformation of the tissue is an important factor in the RFA modelling, which allows the calculation of the force-insertion depth relation as well as the power dissipated in the tissue. In addition, the elastic insertion allows for a more realistic and accurate representation of the cardiac tissue. Finally, our computational results show that the sharp insertion overestimates the tissue temperature rise, hinting for steam pop formation even at low contact force and power ablation protocols.

Acknowledgements

This research was supported by the Basque Government through the BERC 2014-2017 and BERC 2018-2021 program and by Spanish Ministry of Economy and Competitiveness MINECO through BCAM Severo Ochoa excellence accreditations SEV-2013-0323 and SEV-2017-0718,

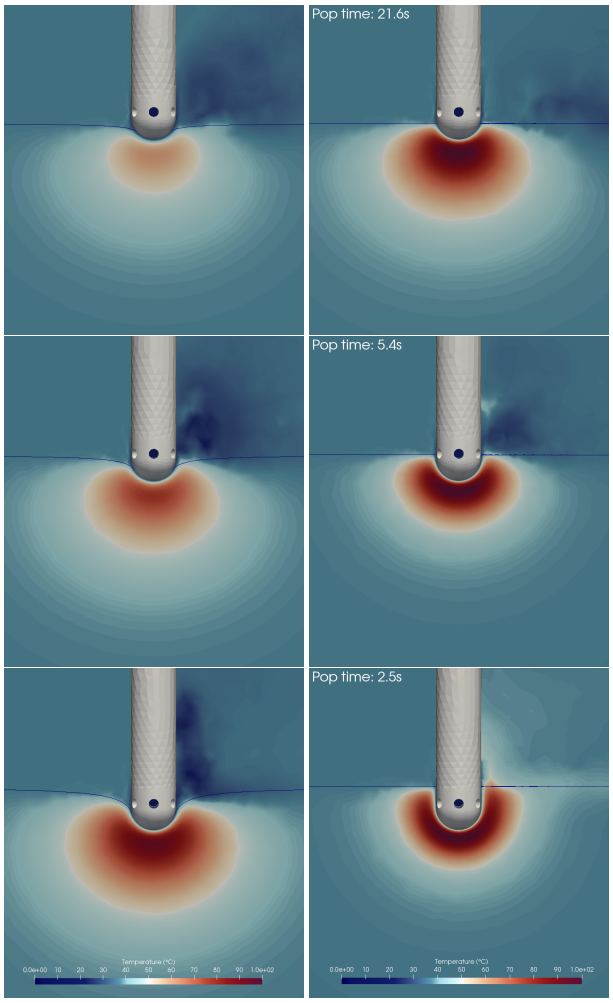


Figure 2. The computational lesions for high blood flow using the elastic insertion (left) and the sharp insertion (right) for $10\,\mathrm{g}$, $20\,\mathrm{g}$ and $40\,\mathrm{g}$ (top to bottom).

and through projects MTM2015-69992-R and MTM2016-76016-R. JJ acknowledges support from project EU H2020 MSO4SC. ML acknowledges the "LaCaixa 2016" PhD Grant.

References

[1] Huang SKS, Wood MA. Catheter Ablation of Cardiac Arrhythmias E-book. Elsevier Health Sciences, 2014.

[2] Berjano EJ. Theoretical modeling for radiofrequency ablation: state-of-the-art and challenges for the future. Biomedical Engineering Online 2006;5(1):24.

[3] González-Suárez A, Berjano E, Guerra JM, Gerardo-Giorda L. Computational modeling of open-irrigated electrodes for radiofrequency cardiac ablation including blood motion-saline flow interaction. PloS One 2016;11(3):e0150356.

[4] Cao H, Speidel MA, Tsai JZ, Van Lysel MS, Vorperian VR, Webster JG. Fem analysis of predicting electrode-myocardium contact from rf cardiac catheter ablation system impedance. IEEE Transactions on Biomedical Engineering 2002;49(6):520–526.

[5] Petras A, Leoni M, Guerra JM, Jansson J, Gerardo-Giorda L. A computational model of open-irrigated radiofrequency ablation including the mechanical properties of the cardiac tissue. International Journal for Numerical Methods in Biomedical Engineering 2018;(submitted).

[6] Sneddon IN. The relation between load and penetration in the axisymmetric boussinesq problem for a punch of arbitrary profile. International Journal of Engineering Science 1965;3(1):47–57.

[7] Wittkampf FH, Nakagawa H. Rf catheter ablation: Lessons on lesions. Pacing and Clinical Electrophysiology 2006; 29(11):1285–1297.

[8] Ribes A, Caremoli C. Salome platform component model for numerical simulation. In Computer Software and Applications Conference, 2007. COMPSAC 2007. 31st Annual International, volume 2. IEEE, 2007; 553–564.

[9] Hoffman J, Jansson J, Jansson N. Fenics-hpc: Automated predictive high-performance finite element computing with applications in aerodynamics. In International Conference on Parallel Processing and Applied Mathematics. Springer, 2015; 356–365.

[10] Ayachit U. The paraview guide: a parallel visualization application. Kitware, Inc., 2015.

[11] Hasgall P, Neufeld E, Gosselin M, Klingenböck A, Kuster N. Itis database for thermal and electromagnetic parameters of biological tissues. Version 30 2015;.

[12] Bhavaraju N, Valvano J. Thermophysical properties of swine myocardium. International Journal of Thermophysics 1999;20(2):665–676.

[13] Urban MW, Pislaru C, Nenadic IZ, Kinnick RR, Greenleaf JF. Measurement of viscoelastic properties of in vivo swine myocardium using lamb wave dispersion ultrasound vibrometry (lduv). IEEE Transactions on Medical Imaging 2013;32(2):247–261.

Address for correspondence:

Argyrios Petras
BCAM - Basque Center for Applied Mathematics
Mazarredo 14, 48009, Bilbao, Basque Country, Spain
apetras@bcamath.org

Systematic characterization of High-Power Short-Duration Ablation: Insight from an advanced virtual model

Short title: Systematic characterization of HPSD ablation

Argyrios PETRAS, PhD[†, a,1], Zoraida MORENO WEIDMANN, MD[†, b], Massimiliano LEONI, MSc[a,c], Jose M GUERRA, MD, PhD[‡, b, §] and Luca GERARDO-GIORDA, PhD[‡, a, 2]

Author Footnotes:

† AP and ZMW are both considered to be first author.

‡ JMG and LGG are both considered to be last author.

[a]BCAM – Basque Center for Applied Mathematics, Alameda Mazarredo 14, 48009 Bilbao, Spain; mleoni@bcamath.org;

[b]Department of Cardiology, Hospital de la Santa Creu i Sant Pau, CIBERCV, Universitat Autonoma de Barcelona, Sant Antoni M Claret 167, 08025 Barcelona, Spain; zmoreno@santpau.cat;

[1]*Present address:* IHU LYRIC-Université de Bordeaux, Avenue du Haut Lévêque, 33600 Pessac, France; argyrios.petras@ihu-liryc.fr

[2] *Present address:* Johannes Kepler University and RICAM, Austrian Academy of Sciences, Altenberger Straße 69, A-4040 Linz, Austria; luca.gerardo-giorda@jku.at

§ ***Corresponding author*****:** Jose M Guerra, MD, PhD. Department of Cardiology, Hospital de la Santa Creu i Sant Pau. Sant Antoni M Claret 167, 08025 Barcelona, Spain. Phone: +34 935565945, Fax: +34 935565603, Email: jguerra@secardiologia.es

Conflicts of interest: Dr Guerra has served as consultant for Biosense Webster, Boston Scientific and Abbott, received speaker fees from Boston Scientific and Abbott, and received a research grant from Abbott. All other authors have no conflicts to disclose.

Word count: 4949

Abstract

Background. High-Power Short-Duration (HPSD) recently emerged as a new approach to radiofrequency (RF) catheter ablation. However, basic and clinical data supporting its effectiveness and safety is still scarce.

Objective. We aim to characterize HPSD with an advanced virtual model, able to assess lesion dimensions and complications in multiple conditions and compare it to standard protocols.

Methods. We evaluate, on both atrium and ventricle, three HPSD protocols (70W/8s, 80W/6s and 90W/4s) through a realistic 3D computational model of power-controlled RF ablation, varying catheter tip design (spherical/cylindrical), contact force, blood flow and saline irrigation. Lesions are defined by the 50ºC isotherm contour. Ablations are deemed safe or complicated by pop (tissue temperature >97ºC) or thrombus (blood temperature >80ºC). We compared HPSD with standards protocols (30-40W/30s). We analyzed the effect of a second HPSD application.

Results. We simulated 432 applications. Most (79%) associated a complication, especially in the atrium. The three HPSD protocols performed similarly in the atrium, while 90W/4s appeared the safest in the ventricle. Low irrigation rate led frequently to charring (72%). HPSD lesions were 40-60% shallower and smaller compared to standards, although featuring similar width. A second HPSD application increased lesions to a size comparable to standards.

Conclusion. HPSD lesions are smaller and shallower than standards but comparable in width, which can be advantageous in the atrium. A second application can produce lesions similar to standards in a shorter time. Despite its narrow safety margin, HPSD seems a valuable new clinical approach.

Keywords: Radiofrequency ablation, High-Power Short-Duration, Computational modeling and simulation, ablation catheter, lesion science

58 ***Introduction***

59 Radiofrequency (RF) catheter ablation has become a cornerstone treatment for
60 cardiac arrhythmias.[1, 2] RF lesions depend mainly on the power delivered to the tissue,
61 catheter stability and optimal contact force (CF) over time.[3, 4] In order to maximize lesion
62 size while minimizing the risk of complications, the maximum power with 4 mm-tip
63 catheters has been limited to 30 to 50 W over 30 to 90 seconds.[3] Nevertheless, catheter
64 stability and optimal CF over time are challenging to keep in a beating heart. This may
65 result in smaller and inconsistent lesions that hamper the creation of effective lines of
66 electrical block and prolong the duration of the procedures.

67 Aimed at overcoming these limitations of conventional protocols, a new RF
68 catheter ablation paradigm has emerged in the last few years. Known as high-power
69 short-duration (HPSD) ablation, these protocols are based on increasing maximum
70 power and markedly shortening the application. An heterogeneity of power/duration
71 settings, ranging from 35 to 60W and 6 to 15 seconds, have been proposed and tested
72 for pulmonary vein isolation.[5-8] Some groups have investigated even more aggressive
73 protocols with power ranging from 70 to 90W during 4 to 8 seconds.[9, 10] Nonetheless,
74 lesion science of HPSD is mostly unknown and basic[10, 11] and clinical data supporting
75 their effectiveness and safety is scarce. In addition, experience on other atrial substrates
76 is limited and lacking on the ventricle.

77 The development of advanced virtual models for RF ablation allows to simulate
78 lesion formation mimicking infinite scenarios. A systematic and realistic evaluation of RF
79 ablation can be performed by changing parameters such as catheter type and shape,
80 ablation protocol, blood and irrigation flow, CF or the cardiac chamber. This will grant an
81 understanding of RF lesion formation and its complications beyond that obtained through
82 classical *in-vitro* and *in-vivo* assessment. This novel research approach proves of
83 particular interest in early stages of new ablation strategies like HPSD.

84 We aim to systematically characterize different HPSD ablation protocols by
85 means of an advanced virtual model developed by our group.[12-15] Lesion dimensions

and complications will be evaluated and compared to those of standard conventional protocols, providing valuable insight in view of the clinical application of HPSD.

Methods

Virtual model

Our computational model of RF catheter ablation simulates a human heart considering fixed conditions, in order to standardize the comparison of the different ablation settings. The mathematical model, based on an *in-vitro* experimental setup,[16] is a three-dimensional computational framework described in previous works of our group.[12-15] It considers the tissue elasticity, the thermoelectric interaction between the electrode and the ablation area, and the convective cooling by the blood flow and the irrigated saline.

According to the thermodynamic principles, heat expansion through the tissue occurs in a radial manner, following a linear temperature-dependent decrease of specific heat (0.42ºC per 1ºC increase) and thermal conductivity (0.05% per 1ºC increase) as well as a linear temperature-dependent increase in electrical conductivity (0.15% per 1ºC increase).[17, 18] The governing equation for the thermal problem is the *Bioheat Equation*.[19]

The model, detailed in the Supplementary Material, is solved by finite elements through our self-developed software in FEniCS-HPC (www.fenics-hpc.org). The computational geometry features about 5,000,000 tetrahedra with a maximum element size of 1 mm and a minimum of 0.01 mm in the area of the electrode, where higher computational accuracy is needed. Simulations were carried out on BCAM in-house cluster *Hipatia*, which features 18 nodes (1 with Nvidia Tesla K40 GPU) for 624 cores with 4TB RAM and Inifiniband network connectivity.

Electrode tip design

We consider two different, commonly used, open-irrigated electrode designs, one with a hemispherical tip and one with a cylindrical tip (7 French, 3.5mm length). Both feature 6 irrigation pores (0.5mm diameter).(Figure 1)

Lesion Assessment

Virtual lesions are identified by the isotherm contour as previously described,[14, 20] considering irreversible tissue damage at 50ºC. The assessment of the lesion dimensions is performed with ParaView.[21] We measure lesion volume, maximum diameter, maximum depth, and depth at the maximum diameter, considering the zero at the undeformed endocardial surface, as described in Figure 1.

Complications Assessment

A steam pop is considered to occur when the tissue temperature exceeds 97ºC. We use a temperature threshold lower than 100ºC to account for the natural fluctuation of water boiling point depending on tissue composition and other physical conditions.[22] A thrombus is assumed to be initiated at a blood temperature of 80ºC.

The occurrence of one of the two complications ends the application, precluding the development of the other. Applications that finish without any complication are deemed safe.

Ablation settings

Three different HPSD ablation protocols are used, 70W/8s, 80W/6s and 90W/4s, delivered in power-control mode without temperature limit. Each of them is simulated on the atrium and the ventricle, varying CF, blood flow velocity and saline irrigation level. All combinations are summarized in Table 1.

In order to simulate different representative cardiac regions, three different blood flow velocities are used: no blood flow (0m/s), as expected in the epicardium or below a valve leaflet; low blood flow (0.1m/s), as expected in a healthy atrium or pulmonary veins; and high blood flow (0.5m/s), as expected at the cavo-tricuspid annulus, the ventricle or outflow tracts.

Additionally, three different saline catheter irrigation rates are simulated for each HPSD protocol: normal, high and very high rate (17ml/min, 30ml/min and 60ml/min, respectively). The blood flow and its interaction with the irrigated saline in the blood compartment are modelled using the incompressible Navier-Stokes equations for fluid dynamics.[19] We assume a room temperature of 22ºC of the irrigation saline.

All virtual applications are performed perpendicularly to the endocardial surface. The lesions are simulated on both atrial and ventricular tissue. Based on clinical data, we assume similar biophysical properties for both cardiac regions, with the sole differentiation of the thickness (6mm in atrium and 12mm in ventricle).[23] Four increasing CF from 5g to 20g are simulated resulting in progressive deformation of the tissue.

Design of the study

The study is structured in three steps:

A. We evaluate the safety of the different HPSD ablation settings and protocols.
B. We assess the efficacy of HPSD protocols identified as the safest in the previous step. For both chambers, we compute HPSD lesion sizes and compare them with those generated by standard applications (30W/30s for the atrium and 30-40W/30s for the ventricle, CF 10 g, irrigation rate 17 ml/min).
C. We analyze the impact of a second HPSD application on top of the first one, in terms of safety and increase in lesion size. We test at increasing time intervals (2s, 4s, …10s) between first and second application. Irrigation rate is kept at

2ml/min between the two applications, while the protocol irrigation rate is restored one second prior to the second application.

Results

A total of 432 RF catheter ablation lesions are simulated by applying the different predefined ablation parameters (catheter tip design, CF, RF power, duration, saline irrigation rate) and different physiological scenarios (blood flow, ablation site).

Safety of HPSD

The main findings are summarized in Figure 2 and Supplementary Table 1. Five conclusions can be highlighted:

a. The safety margin of HPSD ablation is, in general, poor, considering that most of the applications associate some complication. This finding is more notable in the atrium.

b. Low saline flow mostly leads to charring, whereas an increase in the flushing rate carries a higher risk of pop. Similarly, high blood flow also associates an increased number of pops.

c. Almost all applications performed with low CF are free of complications at any HPSD protocol.

d. Different catheter tip designs show varied behaviors. Spherical tips appear relatively safe in both chambers, while cylindrical tips only when used in the ventricle. Nevertheless, cylindrical tips in the ventricle allow the use of larger CF.

e. The complication distribution (thrombus vs pop) is similar for both catheters in the ventricle, whereas in the atrium the spherical tip produces more thrombi and the cylindrical more pops. (Figure 3 and Supplementary Figures 1 and 2)

Efficacy of HPSD

From the previous step, we conclude that the safest setting is represented by CF 5g, irrigation rate 60ml/min, using a spherical tip in the atrium and a cylindrical tip in the ventricle.

The analysis of the different HPSD protocols shows that as power increases and duration diminishes, lesions tend to be narrower and more superficial, resulting in a smaller volume. (Figure 4)

In general, HPSD generates lesions that are smaller than those obtained with standard protocols, especially in terms of depth. This is always the case in the atrium. In the ventricle, HPSD lesions are overall wider but shallower when compared to 30W/30s standard lesions, resulting similar in volume (with the exception of 90W/4s). However, HPSD lesions are consistently smaller when compared to the 40W/30s protocol.

We analyze the effect of increasing CF in the ventricle for the 90W/4s protocol with a cylindrical catheter which was the safest setting observed. At 20g, lesions become comparable to the ones from standard 30W/30s protocols but remain markedly smaller than those from 40W/30s protocols. (Figure 5, Supplementary Figure 3)

Repeated HPSD applications

A second identical HPSD application on top of a first one is simulated after a variable interval. We consider a spherical tip in the atrium and a cylindrical tip in the ventricle, CF of 5g, low blood flow and irrigation rate of 60ml/min, in accordance with the results of the previous step. In general, all lesions increase in size after a repeated application. (Figure 6, Supplementary Table 2) As the interval between applications gets longer, lesion size increase is smaller. Furthermore, as the interval decreases, the risk of complications is higher. For both cardiac chambers, the 80W/6s protocol shows the worst profile, requiring the longest intervals to avoid complications.

In the atrium, increase in width is more pronounced than in depth, being more evident at 90W/4s. Only 70W/8s applications with a 4-second interval reach a final lesion volume

that exceeds that of a lesion performed with a standard protocol (Supplementary Figure 4).

In the ventricle, the lesion increases similarly in depth and width. The best trade-off between safety and lesion increase is reached with a 90W/4s protocol and 6-second interval (77% volume increase).

The final lesion is larger than the standard 30W/30s lesion for any HPSD protocol, especially in terms of width, but generally smaller when compared to those obtained with 40W/30s (Supplementary Figure 4).

Discussion

This is the first study where an advanced virtual model is used to systematically evaluate HPSD ablation. In addition, our innovative model provides a comprehensive insight in the mechanisms of RF ablation, setting the basis for future research in the field.

The major findings of the study are: 1. The safety margin of HPSD is in general poor; 2. Catheter tip design and irrigation rate are critical to ensure safe and efficient applications; 3. HPSD lesions are shallower and smaller compared to standard lesions; 4. A second HPSD application after a suitable time lag increases lesion size.

Safety margin of HPSD

Standard RF protocols currently used are mostly safe.[24, 25] HPSD represents a new approach to RF ablation aimed at reducing ablation time and overcoming some limitations of standard protocols. In agreement with the available published data,[5] our results show that HPSD can be effectively used, although its safety margin is tight. According to our study, the main determinants of complications are the irrigation rate and the catheter tip design, the latter mostly related to the chamber where the RF is applied. To the best of our knowledge, this is the first time that an association between catheter tip shape, anatomical location of the ablation and safety is shown. These findings

suggest that the clinical use of HPSD should be implemented with caution and that additional safety measures should be considered.

Influence of catheter tip design

Catheter tip shape has a crucial impact on the safety profile of HPSD. We show that while a spherical catheter tip provides the best trade-off between safety and efficacy in the atrium, the opposite occurs in the ventricle.

A possible explanation for this observation can be found in the physics of RF ablation. The power delivered to the tissue increases proportionally to the percentage of the catheter tip that is in contact with the wall. Due to tissue deformation, the contact surface is smaller for spherical tips at low CF (<10g) and for cylindrical tips at high CF (>10g). (Supplementary Figure 5) As a result, at low CF the spherical tip delivers less power than the cylindrical ones while at high CF the opposite occurs. Heat dissipation in the thinner atrial wall is lower,[17] limiting the power that can be safely delivered to this chamber. In this case, only the spherical tip at low CF appears to remain within the safety margins. Contrarily, the thicker ventricular wall allows higher power delivery and the cylindrical tip appears more effective and safer.

Although few studies have analyzed the role of electrode tip shape on standard ablations, a similar pattern was already observed with standard protocols in previous *in vitro* studies[16, 26] comparing two commercially available catheters, ThermoCool® (Biosense Webster, cylindrical design) and CoolPath® (St Jude Medical, spherical design).

Catheter tip design and effective cooling

As previously demonstrated,[27, 28] the use of catheter cooling systems allows larger power delivery preventing char and thrombus formation, particularly in areas with poor blood flow. Based on our results, a considerably high flushing rate (>30ml/min) is necessary to avoid charring, which can be a limitation for heart failure patients. The spherical tip also

led more frequently to charring, especially in the atrium, possibly related to a larger surface of the tip in contact with blood than with tissue.

Charring mostly appears in our model at the interface between the metallic tip and the rest of the catheter body. We replicate a cooling system based on six distal coplanar pores, with saline flowing out perpendicularly to the electrode surface. This is probably not the most suitable configuration since the electrode-body interface is not directly cooled down. Additionally, an increase in the irrigation rate does not help: the higher flow velocity just pushes the saline farther away from the tip with no significant impact on the cooling of the upper part of the electrode. This advocates for a more efficient cooling system, which could be achieved by either adding more pores distributed along the entire metallic tip, or by forcing a directionality in the irrigation flow towards the upper part of the electrode.[10, 11]

HPSD lesion size and shape

In the line of the experimental data reported in HPSD *in-vitro* experiments,[10, 11] HPSD lesions are wider than they are deep and, in general, smaller than those of standard protocols. Moreover, increases in power, combined with shorter duration of application, entail a decrease in depth accompanied by minor changes in width. This can be explained by the distinctive biophysics of heat transfer in HPSD protocols, since lesion generation is mainly based on resistive heating, while conductive heating has only a marginal role due to the short duration of the application.[17]

This feature may prove advantageous in the atrium, allowing to treat the arrhythmogenic substrate in its thin wall while avoiding complications in its vicinity. However, it may be insufficient to reach deep targets as in the ventricle or for mitral annular lines.[8]

HPSD protocols comparison

As power is increased and duration reduced shallower lesions are generated with minimal change in width. In terms of safety, we did not find any difference between HPSD

protocols in the atrium. In contrast, in the ventricle, the 90W/4s protocol represented the best power/duration balance to ensure safe lesions while the 80W/6s appeared to be the poorest. The 70W/8s protocol offered a safe margin, but only at moderate CF.

Leshem et al.[10] also identified the 90W/4s as the best power/duration combination, allowing them to safely apply CF as high as 40g. Differently to our approach, these authors used a highly sensitive and efficient temperature-control catheter with a rapid power down-regulation at temperature >65ºC and a cylindrical tip design with 64-pores, which can explain the wider safety margin with respect to CF.

Effect of a second application

We observed a substantial increase in lesion size after a second identical HPSD application, in opposition to what is expected to occur with standard protocols.[29] According to thermodynamics, heat transfer within the tissue continues until thermal equilibrium is reached. This process is estimated to last 45 to 60 seconds which is close to the end of the application in standard protocols. The short duration of HPSD protocols allows to exploit this thermal latency period by rapidly adding a second application to increase the lesion size.

We analyzed different intervals between applications. The optimal interval was a trade-off between efficacy and safety: the shorter the interval the larger the lesion size gain but, at the same time, the higher the risk of pops.

Two additional findings should be highlighted. In the atrium, the gain in lesion size was more evident in width than in depth, which would potentially maximize the extension of atrial lesions, while preserving neighboring structures. In the ventricle, the second application achieved lesions comparable in size to standard ones, notably reducing the ablation time.

Limitations

Our advanced virtual model includes several variables which influence the biophysics of RF ablation, making it comparable to *in-vitro* and *in-vivo* models. However, we still miss some factors such as the tissue heterogeneities and anisotropies. Additionally, the model assumes constancy of other variables such as catheter position and orientation, CF or blood flow. However, in our opinion, these limitations do not have a significant impact on the global results of the study.

We chose three HPSD protocols with powers above 70W to focus on lesions basically derived from resistive heating, whose biophysics is mostly unknown. Lower powers and longer duration could be considered but they are beyond the scope of the study.

Our model assumes a power-controlled system, fully delivering the programmed power without an upper temperature limit. Adding a self-down-regulation of the energy at a given tissue or blood temperature would probably avoid some complications observed in our simulations, but would prevent a deeper understanding of the behavior of this type of energy delivery.

The simulated applications were interrupted when the blood temperature or the tissue temperature reached complication thresholds. Our model uses a very simple 6 pores irrigation system prone to charring. The question lingers whether by using a more effective irrigation system it could have been avoided and if so, whether pops would have occurred had the applications continued. Although this could change the safety profile observed, this analysis goes beyond the scope of this paper and warrants further investigation.

Conclusions

HPSD ablation generates lesions that are both shallower and smaller in volume than those of standard protocols. In the atrium, these characteristics can be exploited to reach the arrhythmogenic substrate while preserving the neighboring structures. In the ventricle, a second application can create lesions comparable to the ones obtained with standard protocols, but in a shorter time. The safety margin of HPSD is narrow, and

therefore it should be implemented with caution and probably using *ad hoc* designed RF systems. Our advanced virtual model, providing a deep insight on the physiology of RF ablation, can play a key role in the future development of new technologies and protocols.

Acknowledgements / Funding:

This research was supported by the Basque Government through the BERC 2018-2021 program, and the Spanish State Research Agency (AEI) through the BCAM Severo Ochoa excellence accreditation SEV-2017-0718. LGG was also funded jointly by AEI and Feder through the grant RTI2018-093416-B-I00 MULTIQUANT.

ML acknowledges the "LaCaixa 2016" PhD Grant.

REFERENCES

1. Calkins H, Hindricks G, Cappato R, et al. 2017 HRS/EHRA/ECAS/APHRS/SOLAECE expert consensus statement on catheter and surgical ablation of atrial fibrillation. Heart Rhythm 2017;14:e275-e444.

2. Cronin EM, Bogun FM, Maury P, et al. 2019 HRS/EHRA/APHRS/LAHRS expert consensus statement on catheter ablation of ventricular arrhythmias. Heart Rhythm 2020;17:e2-e154.

3. Wittkampf FH, Nakagawa H. RF catheter ablation: Lessons on lesions. Pacing Clin Electrophysiol 2006;29:1285-1297.

4. Kuck K, Brugada J, Albenque J. Cryoballoon or Radiofrequency Ablation for Atrial Fibrillation. N Engl J Med 2016;375:1100-1101.

5. Vassallo F, Cunha C, Serpa E, et al. Comparison of high-power short-duration (HPSD) ablation of atrial fibrillation using a contact force-sensing catheter and conventional technique: Initial results. J Cardiovasc Electrophysiol 2019;30:1877-1883.

6. Winkle RA, Mohanty S, Patrawala RA, et al. Low complication rates using high power (45-50 W) for short duration for atrial fibrillation ablations. Heart Rhythm 2019;16:165-169.

7. Castrejon-Castrejon S, Martinez Cossiani M, Ortega Molina M, et al. Feasibility and safety of pulmonary vein isolation by high-power short-duration radiofrequency application: short-term results of the POWER-FAST PILOT study. J Interv Card Electrophysiol 2020;57:57-65.

8. Yavin HD, Leshem E, Shapira-Daniels A, et al. Impact of High-Power Short-Duration Radiofrequency Ablation on Long-Term Lesion Durability for Atrial Fibrillation Ablation. JACC Clin Electrophysiol 2020;6:973-985.

9. Bourier F, Duchateau J, Vlachos K, et al. High-power short-duration versus standard radiofrequency ablation: Insights on lesion metrics. J Cardiovasc Electrophysiol 2018;29:1570-1575.

10. Leshem E, Zilberman I, Tschabrunn CM, et al. High-Power and Short-Duration Ablation for Pulmonary Vein Isolation: Biophysical Characterization. JACC Clin Electrophysiol 2018;4:467-479.
11. Rozen G, Ptaszek LM, Zilberman I, et al. Safety and efficacy of delivering high-power short-duration radiofrequency ablation lesions utilizing a novel temperature sensing technology. Europace 2018;20:f444-f450.
12. Petras A, Leoni M, Guerra J, Jansson J, Gerardo-Giorda L. Effect of Tissue Elasticity in Cardiac Radiofrequency Catheter Ablation Models. 2018 Computing in Cardiology Conference (CinC). Maastricht, Netherlands, 2018; 1-4. doi: 10.22489/CinC.2018.035
13. Petras A, Leoni M, Guerra JM, Jansson J, Gerardo-Giorda L. Tissue Drives Lesion: Computational Evidence of Interspecies Variability in Cardiac Radiofrequency Ablation. In: Coudière Y, Ozenne V, Vigmond E, Zemzemi N, eds. Functional Imaging and Modeling of the Heart. FIMH 2019. Lecture Notes in Computer Science. Vol 11504: Springer; 2019:139-146.
14. Petras A, Leoni M, Guerra JM, Jansson J, Gerardo-Giorda L. A computational model of open-irrigated radiofrequency catheter ablation accounting for mechanical properties of the cardiac tissue. Int J Numer Method Biomed Eng 2019;35:e3232.
15. Petras A, Echeverria Ferrero M, Leoni M, Guerra JM, Jansson J, Gerardo-Giorda L. Stay on the safe side: in-silico assessment of ablation protocols to prevent steam pops during radiofrequency ablation. European Heart Journal 2019;40, Supplement_1
16. Guerra JM, Jorge E, Raga S, et al. Effects of open-irrigated radiofrequency ablation catheter design on lesion formation and complications: in vitro comparison of 6 different devices. J Cardiovasc Electrophysiol 2013;24:1157-1162.
17. Haines DE, Watson DD. Tissue heating during radiofrequency catheter ablation: a thermodynamic model and observations in isolated perfused and superfused canine right ventricular free wall. Pacing Clin Electrophysiol 1989;12:962-976.

18. Schutt D, Berjano EJ, Haemmerich D. Effect of electrode thermal conductivity in cardiac radiofrequency catheter ablation: a computational modeling study. Int J Hyperthermia 2009;25:99-107.
19. Abraham JP, Sparrow EM. A thermal-ablation bioheat model including liquid-to-vapor phase change, pressure- and necrosis-dependent perfusion, and moisture-dependent properties. International Journal of Heat and Mass Transfer 2007;50:2537-2544.
20. Gonzalez-Suarez A, Berjano E, Guerra JM, Gerardo-Giorda L. Computational Modeling of Open-Irrigated Electrodes for Radiofrequency Cardiac Ablation Including Blood Motion-Saline Flow Interaction. PLoS One 2016;11:e0150356.
21. Ayachit U. The ParaView Guide: A Parallel Visualization Application. United States of America: Kitware Incorporated; 2015.
22. Thompson N, Lustgarten D, Mason B, et al. The relationship between surface temperature, tissue temperature, microbubble formation, and steam pops. Pacing Clin Electrophysiol 2009;32:833-841.
23. Cabrera JA, Sanchez-Quintana D. Cardiac anatomy: what the electrophysiologist needs to know. Heart 2013;99:417-431.
24. Kuck KH, Brugada J, Furnkranz A, et al. Cryoballoon or Radiofrequency Ablation for Paroxysmal Atrial Fibrillation. N Engl J Med 2016;374:2235-2245.
25. Taghji P, El Haddad M, Phlips T, et al. Evaluation of a Strategy Aiming to Enclose the Pulmonary Veins With Contiguous and Optimized Radiofrequency Lesions in Paroxysmal Atrial Fibrillation: A Pilot Study. JACC Clin Electrophysiol 2018;4:99-108.
26. Moreno J, Quintanilla JG, Molina-Morua R, et al. Morphological and thermodynamic comparison of the lesions created by 4 open-irrigated catheters in 2 experimental models. J Cardiovasc Electrophysiol 2014;25:1391-1399.
27. Yokoyama K, Nakagawa H, Wittkampf FH, Pitha JV, Lazzara R, Jackman WM. Comparison of electrode cooling between internal and open irrigation in radiofrequency

ablation lesion depth and incidence of thrombus and steam pop. Circulation 2006;113:11-19.

28. Everett THt, Lee KW, Wilson EE, Guerra JM, Varosy PD, Olgin JE. Safety profiles and lesion size of different radiofrequency ablation technologies: a comparison of large tip, open and closed irrigation catheters. J Cardiovasc Electrophysiol 2009;20:325-335.

29. Wittkampf FH, Nakagawa H, Yamanashi WS, Imai S, Jackman WM. Thermal latency in radiofrequency ablation. Circulation 1996;93:1083-1086.

Table 1. Summary of the different ablation settings and physiological scenarios considered in our simulations combining catheter tip, contact force, power, duration, irrigation rate, blood flow and cardiac chamber.

Ablation settings		
	Catheter tip	Spherical
		Cylindrical
	Contact force	5g
		10g
		15g
		20g
	Power/Duration	70W/8s
		80W/6s
		90W/4s
	Irrigation rate	17mL/min
		30mL/min
		60mL/min
Physiological scenarios		
	Blood flow	0.0m/s
		0.1m/s
		0.5m/s
	Ablation site	Atrium
		Ventricle

Figure 1. Representation of the virtual lesion created with the two considered catheter designs. (A) Ablation performed with a spherical catheter tip and (B) with a cylindrical catheter tip. The area of irreversible tissue lesion is delimited by the blue line, corresponding to a tissue temperature ≥50ºC. Lesion measurements: maximum depth (D), depth at the maximum diameter (DW), maximum diameter (W). Measurements are taken from the undeformed surface identified by the dashed line.

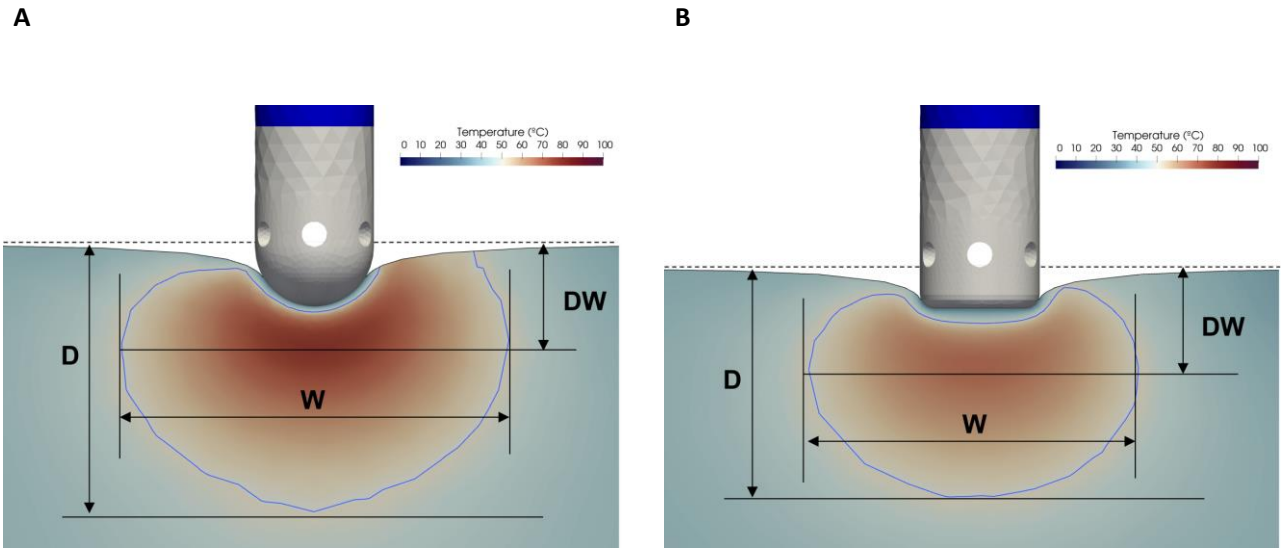


Figure 2. Safety outcomes of HPSD protocols according to the different parameters described in Table 1. HBF = High Blood Flow (0.5m/s); LBF = Low Blood Flow (0.1m/s); NBF = No Blood Flow (0.0m/s).

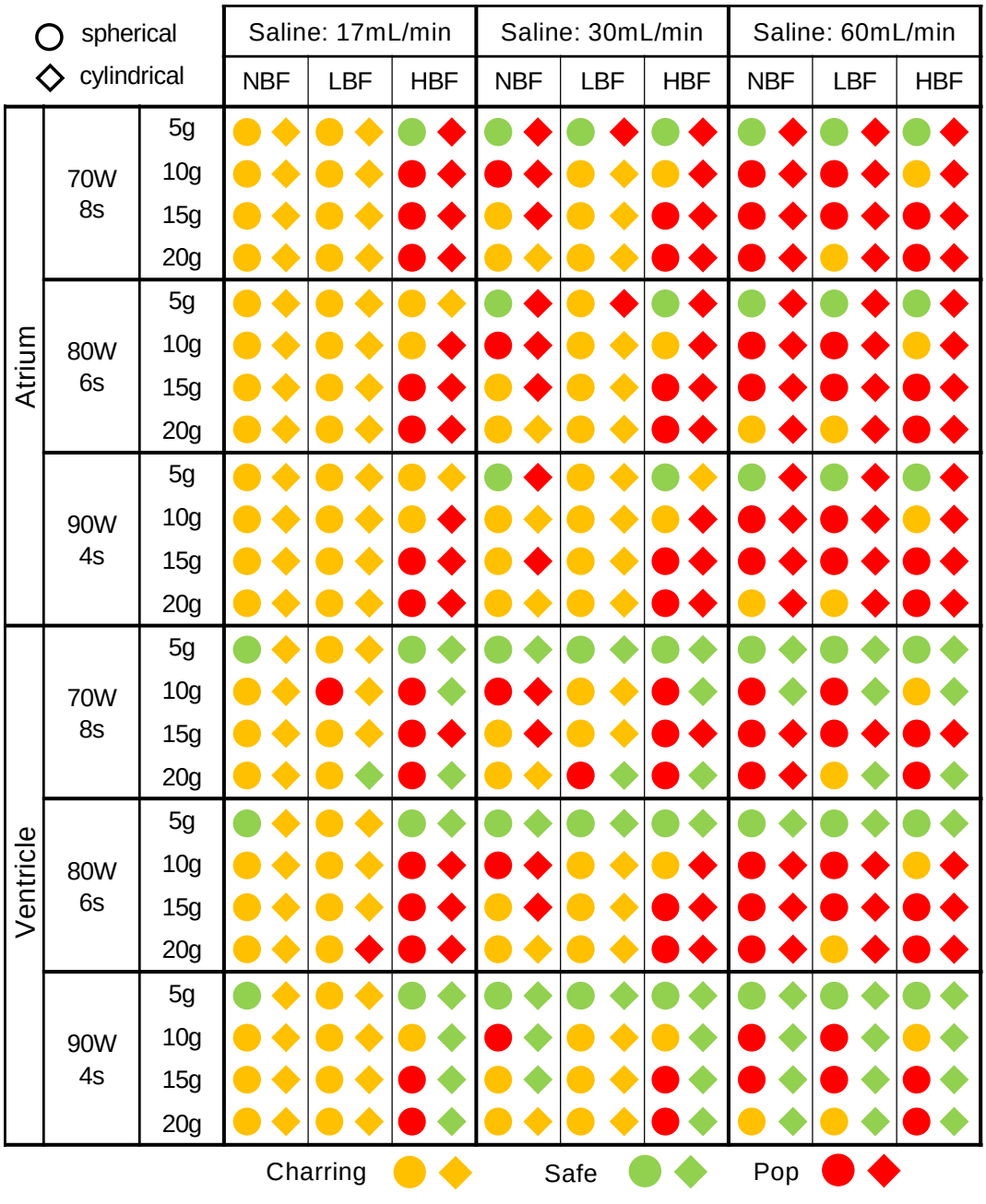


Figure 3. Overall safety comparison of the two different catheter tip designs.

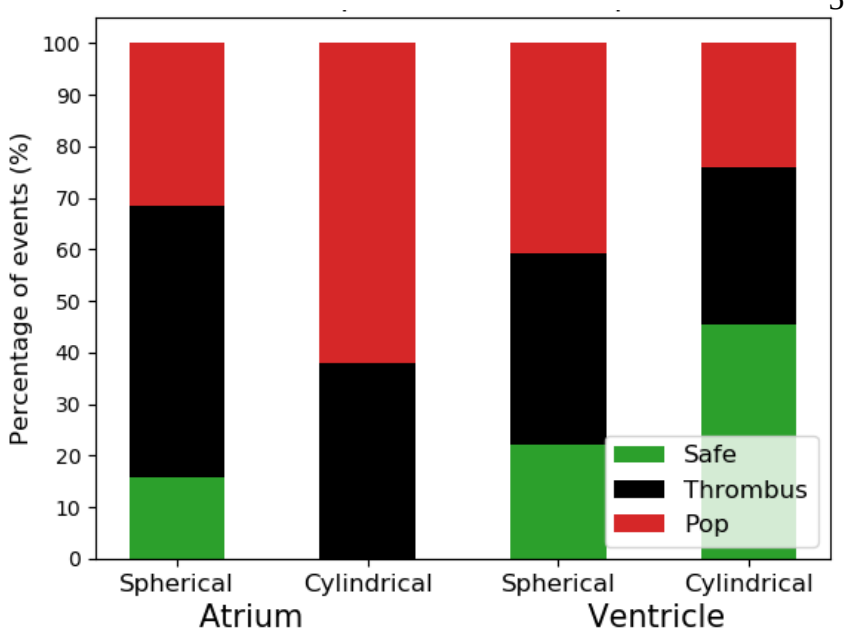
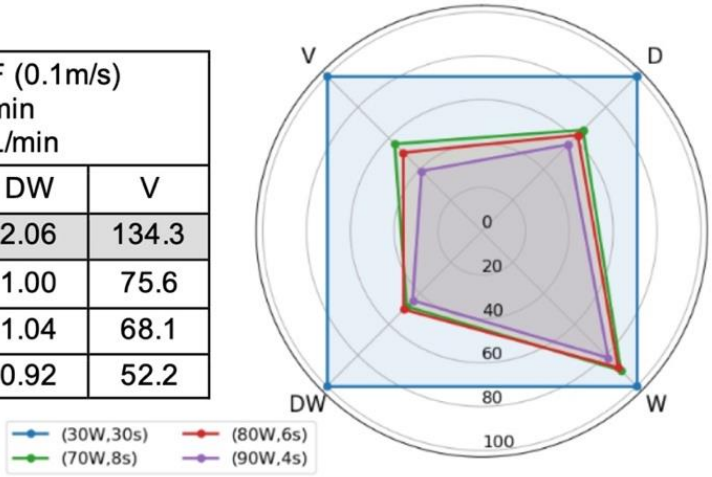


Figure 4. Comparison between HPSD and standard ablation protocols in atrium (A) and ventricle (B). Radar charts show the percentage comparison against a reference standard protocol (30W/30s in the atrium and 40W/30s in the ventricle). The associated tables present the absolute values (standard protocols are highlighted in grey). D = lesion depth, DW = Depth at maximum width, V = volume, W = lesion width.

A.

Spherical tip, Atrium, LBF (0.1m/s) HPSD: 5g, 60mL/min Standard: 10g,17mL/min				
Protocol	D	W	DW	V
30W / 30s	5.31	7.59	2.06	134.3
70W / 8s	3.47	6.82	1.00	75.6
80W / 6s	3.29	6.68	1.04	68.1
90W / 4s	2.97	6.20	0.92	52.2



B.

Cylindrical tip, Ventricle, LBF (0.1m/s) HPSD: 5g, 60mL/min, Standard: 10g,17mL/min				
Protocol	D	W	DW	V
30W / 30s	4.53	6.49	2.06	84.8
40W / 30s	5.30	8.11	2.34	166.1
70W / 8s	3.68	7.10	1.01	88.4
80W / 6s	3.49	6.96	0.32	79.2
90W / 4s	3.07	6.56	0.64	60.0

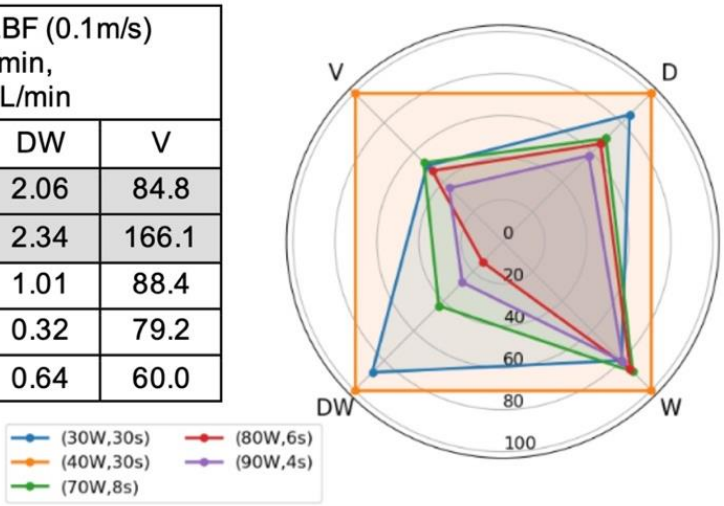
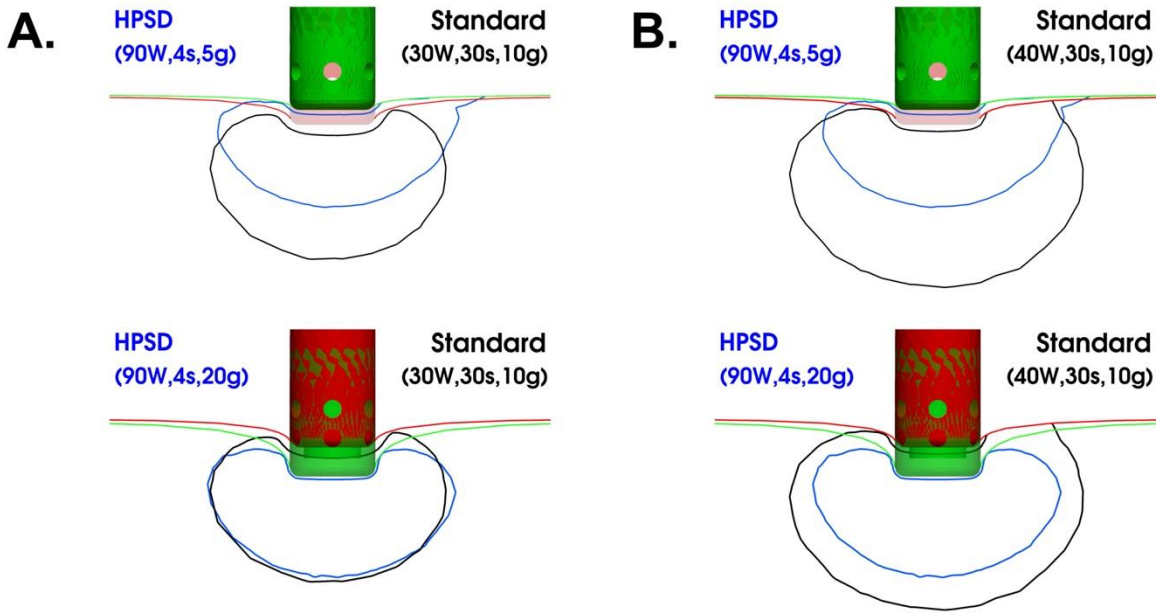


Figure 5. Lesion comparison between 90W/4s and standard protocols in the ventricle at different CF. The table collects the absolute values. 5A shows the comparison of the 90W/4s against standard 30W/30s protocol at CF 5g and 20g; 5B shows the comparison of the 90W/4s against standard 40W/30s protocol at CF 5g and 20g.



Cylindrical, Ventricle, LBF (0.1m/s), HPSD: 90W / 4s, 60mL/min, Standard: 10g,17mL/min				
Protocol	D	W	DW	V
30W / 30s	4.53	6.49	2.06	84.8
40W / 30s	5.30	8.11	2.34	166.1
5g	3.07	6.56	0.64	60.0
10g	3.61	6.95	1.15	73.9
15g	4.02	6.83	1.53	78.3
20g	4.38	6.75	2.00	80.1

Figure 6. Effect on lesion size from a second HPSD application on top of a first one as a function of the time interval between the two. For the second ablation, the same protocol of the first ablation is considered. Ablations are performed with spherical tip in the atrium and cylindrical tip in the ventricle, at CF 5g, blood flow 0.1m/s and 60ml/min irrigation rate. During the pause a 2ml/min saline inlet is considered and the high saline flow is restored 1s before the second ablation. The percentage increase in depth, width and volume is represented as bar charts for both atrium (A) and ventricle (B). Striped bars indicate a pop occurrence during the second application.